Week 1 Assignment

Ben Arancibia

June 11, 2015

HA 2.1 For each of the following series (from the fma package), make a graph of the data. If transforming seems appropriate, do so and describe the effect.

```
library(fma)
```

```
## Loading required package: tseries
## Loading required package: forecast
```

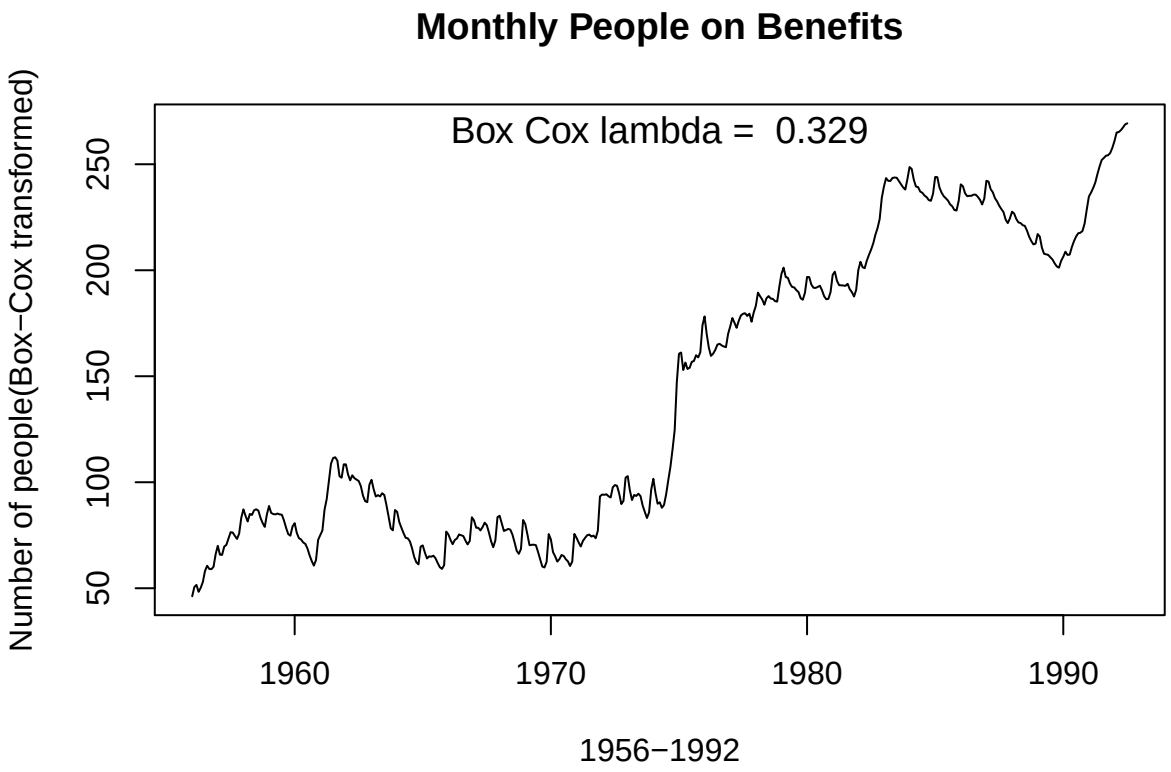
```
## Warning: package 'forecast' was built under R version 3.1.3
```

```
## Loading required package: zoo
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
##
## Loading required package: timeDate
## This is forecast 6.1
```

```
library(mlbench)
library(lattice)
```

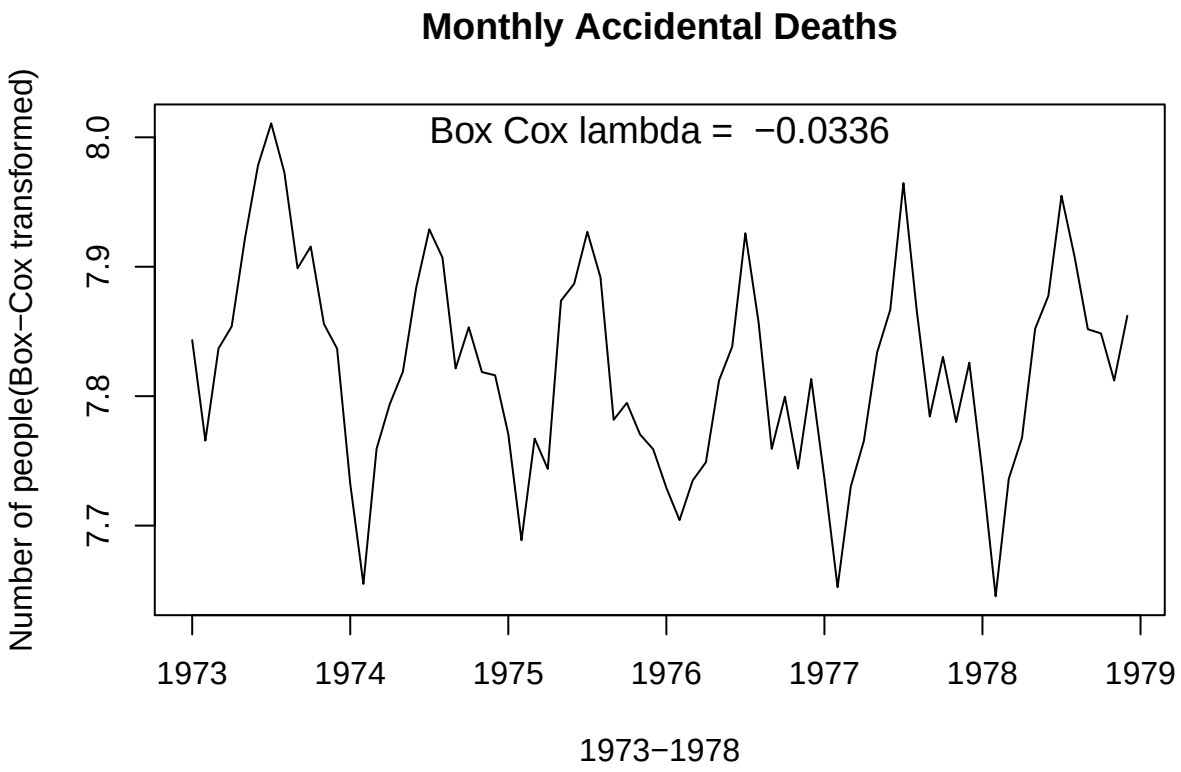
a) Monthly total of people on unemployed benefits in Australia (January 1956–July 1992).

```
lambda.benefits <- BoxCox.lambda(dole)
plot(BoxCox(dole, lambda.benefits), main="Monthly People on Benefits",
     xlab="1956-1992", ylab="Number of people(Box-Cox transformed)")
title(main=paste("Box Cox lambda = ", signif(lambda.benefits, digits=3)), font.main=8, line=-1)
```



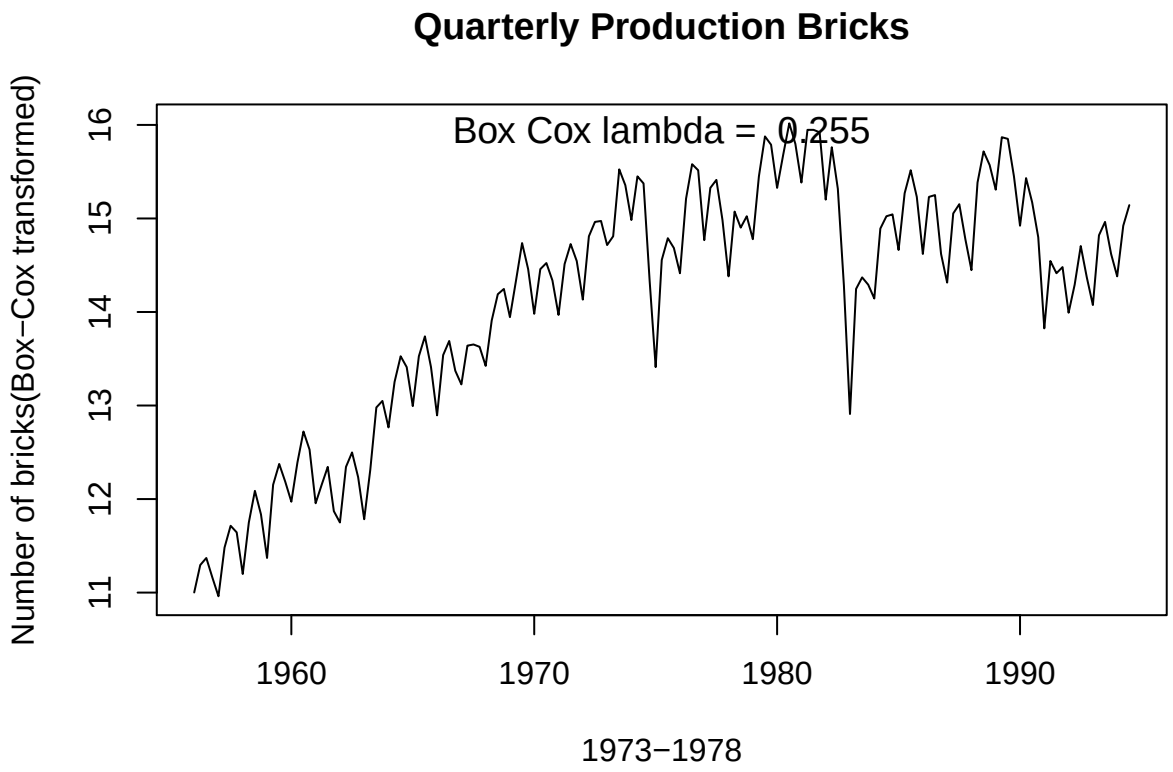
b) Monthly total of accidental deaths in the United States (January 1973–December 1978).

```
lambda.deaths <- BoxCox.lambda(usdeaths)
plot(BoxCox(usdeaths, lambda.deaths), main="Monthly Accidental Deaths",
     xlab="1973-1978", ylab="Number of people(Box-Cox transformed)")
title(main=paste("Box Cox lambda = ", signif(lambda.deaths, digits=3)), font.main=8, line=-1)
```



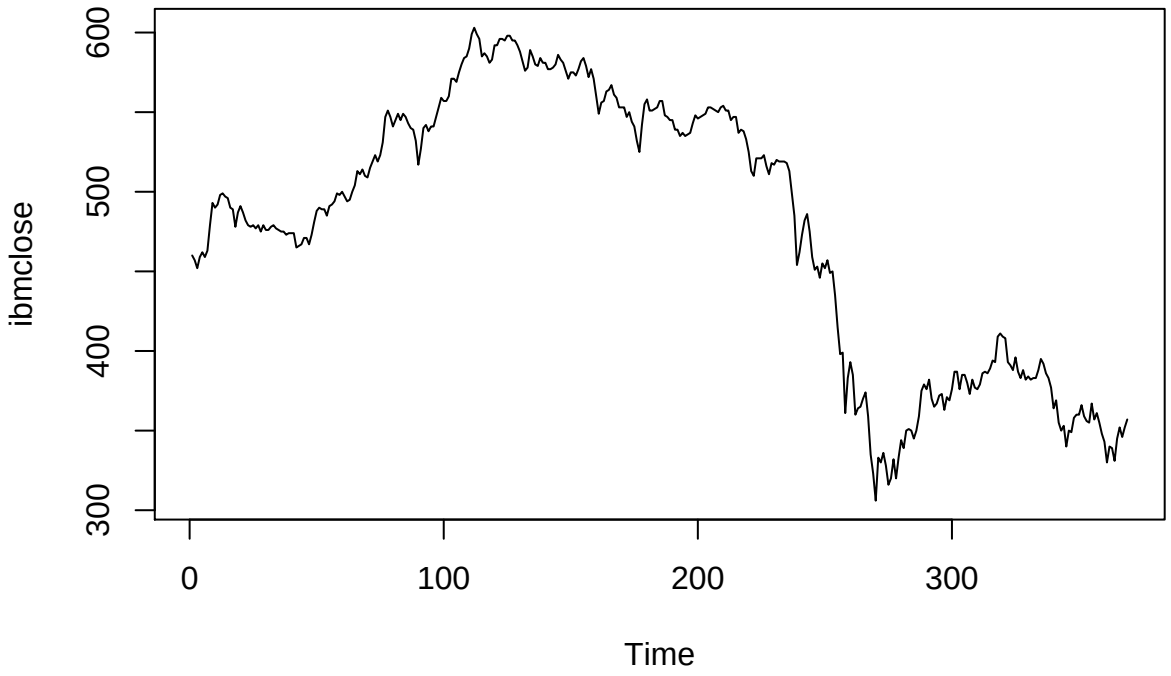
c) Quarterly production of bricks (in millions of units) at Portland, Australia (March 1956–September 1994).

```
lambda.bricks <- BoxCox.lambda(bricksq)
plot(BoxCox(bricksq, lambda.bricks), main="Quarterly Production Bricks",
     xlab="1973-1978", ylab="Number of bricks(Box-Cox transformed)")
title(main=paste("Box Cox lambda = ", signif(lambda.bricks, digits=3)), font.main=8, line=-1)
```

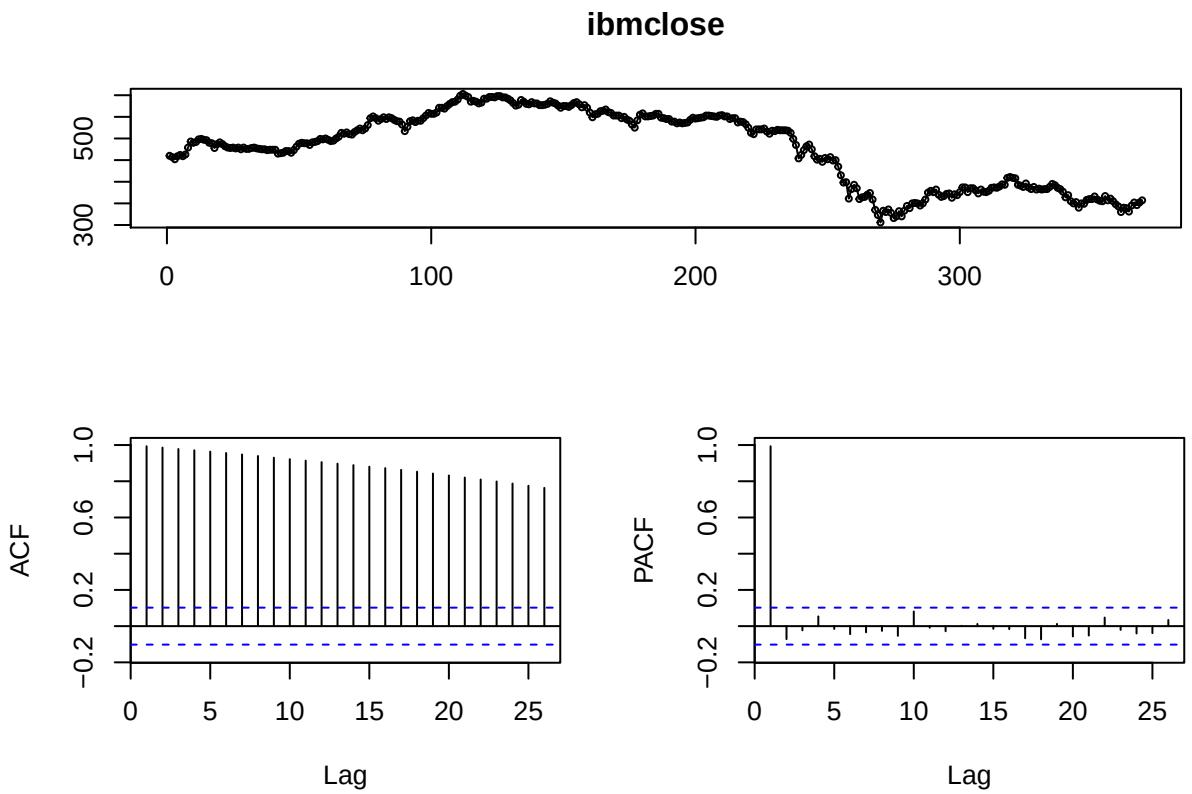


HA 2.3 Consider the daily closing IBM stock prices (data set ibmclose). a)Produce some plots of the data in order to become familiar with it.

```
plot(ibmclose)
```



```
tsdisplay(ibmclose)
```



b)Split the data into a training set of 300 observations and a test set of 69 observations.

```
training <- ibmclose[1:300]
test <- ibmclose[301:369]
```

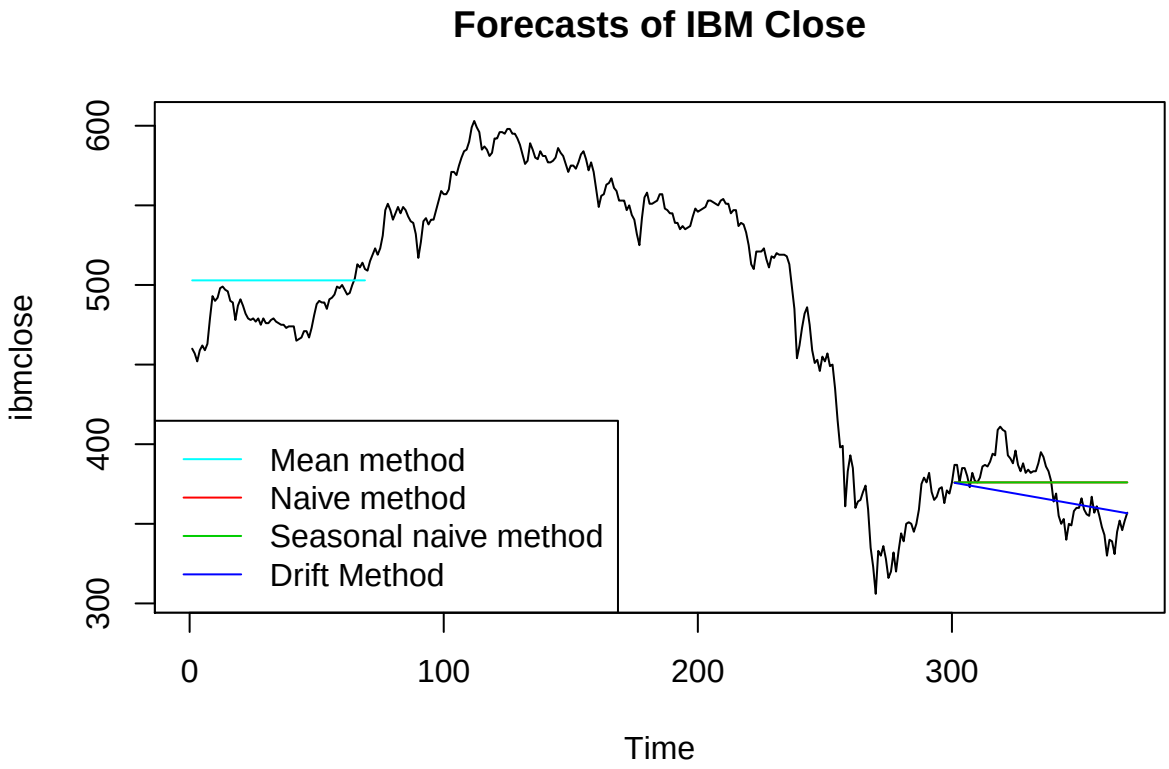
c)Try various benchmark methods to forecast the training set and compare the results on the test set. Which method did best?

```
ibm2 <- window(training)
ibmfit <- meanf(ibm2, h=69)
ibmfit2 <- naive(ibm2, h=69)
ibmfit3 <- snaive(ibm2, h=69)
ibmfit4 <- rwf(ibm2, h=69, drift=TRUE)

plot(ibmclose, plot.conf=FALSE, main="Forecasts of IBM Close")
```

```
## Warning in plot.window(xlim, ylim, log, ...): "plot.conf" is not a
## graphical parameter

## Warning in title(main = main, xlab = xlab, ylab = ylab, ...): "plot.conf"
## is not a graphical parameter

## Warning in axis(1, ...): "plot.conf" is not a graphical parameter

## Warning in axis(2, ...): "plot.conf" is not a graphical parameter
```

```
## Warning in box(...): "plot.conf" is not a graphical parameter
```

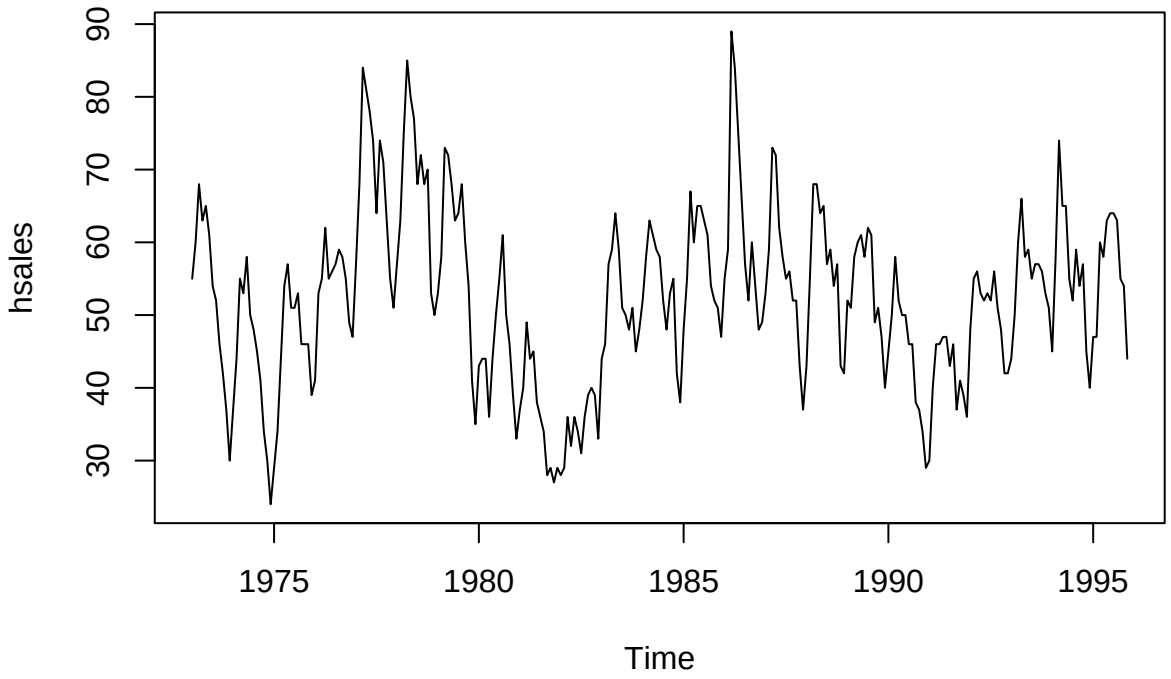
```
lines(ibmfit$mean,col=5)
lines(ibmfit2$mean,col=2)
lines(ibmfit3$mean,col=3)
lines(ibmfit4$mean,col=4)
legend("bottomleft",lty=1,col=c(5,2,3,4),
legend=c("Mean method","Naive method","Seasonal naive method", "Drift Method"))
```



The Drift Method did the best. One thing to note is that when I plot the seasonal naive method and the naive method, they are on top of each other. It makes it look like one did not plot, but one just obscures the other.

HA 2.4 Consider the sales of new one-family houses in the USA, Jan 1973 – Nov 1995 (data set hsales). a) Produce some plots of the data in order to become familiar with it.

```
plot(hsales)
```

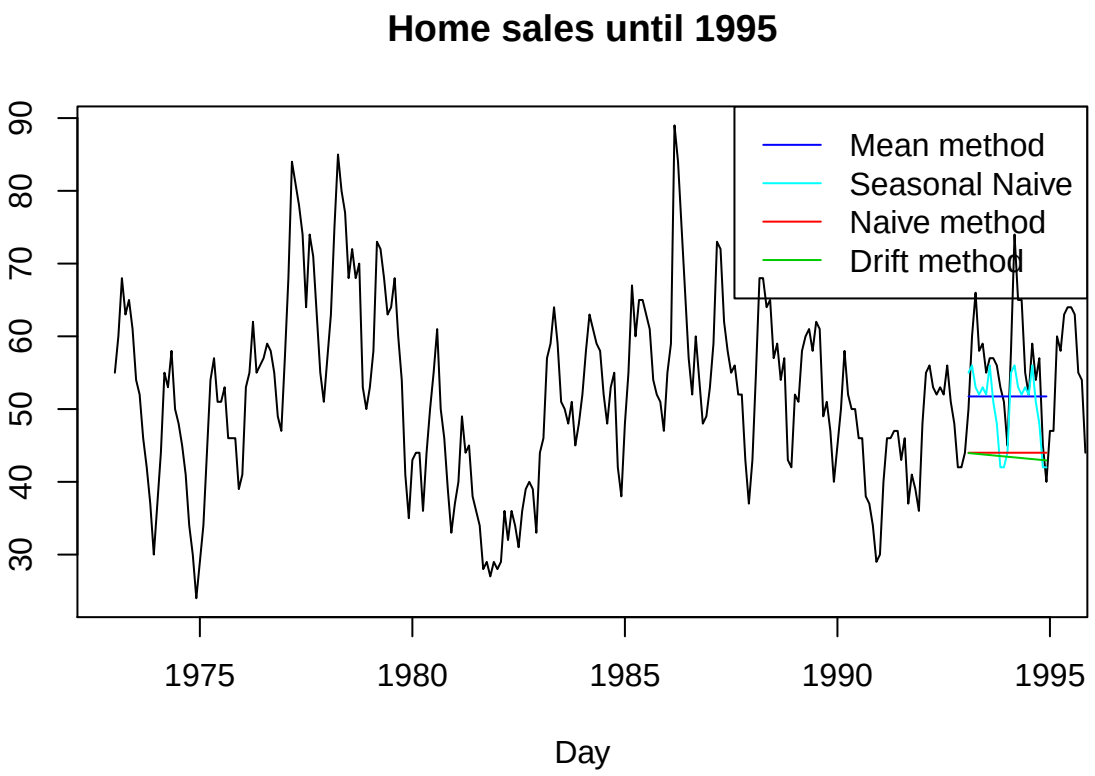


b)Split the hsales data set into a training set and a test set, where the test set is the last two years of data.

```
training2 <- hsales[1:252]
test2 <- hsales[253:275]
```

c)Try various benchmark methods to forecast the training set and compare the results on the test set. Which method did best?

```
hs2 <- window(hsales, end=1993)
plot(hsales,main="Home sales until 1995",
  ylab="",xlab="Day",xlim=c(1973,1995))
lines(meanf(hs2,h=23)$mean,col=4)
lines(snaive(hs2, h=23)$mean,col=5)
lines(rwf(hs2,h=23)$mean,col=2)
lines(rwf(hs2,drift=TRUE,h=23)$mean,col=3)
legend("topright",lty=1,col=c(4, 5 ,2 ,3),legend=c("Mean method","Seasonal Naive","Naive method","Drift
```



The seasonal naive forecast did the best in predicting the Home sales.

```
data(Soybean)
```

KJ 3.2

a) Investigate the frequency distributions for the categorical predictors. Are any of the distributions degenerate in the ways disscused in chapter 3?

```
str(Soybean)
```

```
## 'data.frame':    683 obs. of  36 variables:
##  $ Class          : Factor w/ 19 levels "2-4-d-injury",..: 11 11 11 11 11 11 11 11 11 11 ...
##  $ date           : Factor w/ 7 levels "0","1","2","3",..: 7 5 4 4 7 6 6 5 7 5 ...
##  $ plant.stand    : Ord.factor w/ 2 levels "0"<"1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ precip         : Ord.factor w/ 3 levels "0"<"1"<"2": 3 3 3 3 3 3 3 3 3 3 ...
##  $ temp           : Ord.factor w/ 3 levels "0"<"1"<"2": 2 2 2 2 2 2 2 2 2 2 ...
##  $ hail           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 1 1 ...
##  $ crop.hist      : Factor w/ 4 levels "0","1","2","3": 2 3 2 2 3 4 3 2 4 3 ...
##  $ area.dam       : Factor w/ 4 levels "0","1","2","3": 2 1 1 1 1 1 1 1 1 1 ...
##  $ sever          : Factor w/ 3 levels "0","1","2": 2 3 3 3 2 2 2 2 2 3 ...
##  $ seed.tmt       : Factor w/ 3 levels "0","1","2": 1 2 2 1 1 1 2 1 2 1 ...
##  $ germ           : Ord.factor w/ 3 levels "0"<"1"<"2": 1 2 3 2 3 2 1 3 2 3 ...
##  $ plant.growth   : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ leaves         : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ leaf.halo      : Factor w/ 3 levels "0","1","2": 1 1 1 1 1 1 1 1 1 1 ...
```

```
##  $ leaf.marg      : Factor w/ 3 levels "0","1","2": 3 3 3 3 3 3 3 3 3 3 ...
##  $ leaf.size      : Ord.factor w/ 3 levels "0"<"1"<"2": 3 3 3 3 3 3 3 3 3 3 ...
##  $ leaf.shread    : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ leaf.malf      : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ leaf.mild      : Factor w/ 3 levels "0","1","2": 1 1 1 1 1 1 1 1 1 1 ...
##  $ stem           : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ lodging        : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 2 1 1 1 ...
##  $ stem.cankers   : Factor w/ 4 levels "0","1","2","3": 4 4 4 4 4 4 4 4 4 4 ...
##  $ canker.lesion  : Factor w/ 4 levels "0","1","2","3": 2 2 1 1 2 1 2 2 2 2 ...
##  $ fruiting.bodies: Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ ext.decay      : Factor w/ 3 levels "0","1","2": 2 2 2 2 2 2 2 2 2 2 ...
##  $ mycelium       : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ int.discolor   : Factor w/ 3 levels "0","1","2": 1 1 1 1 1 1 1 1 1 1 ...
##  $ sclerotia      : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ fruit.pods     : Factor w/ 4 levels "0","1","2","3": 1 1 1 1 1 1 1 1 1 1 ...
##  $ fruit.spots    : Factor w/ 4 levels "0","1","2","4": 4 4 4 4 4 4 4 4 4 4 ...
##  $ seed           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ mold.growth    : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ seed.discolor  : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ seed.size      : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ shriveling     : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ roots          : Factor w/ 3 levels "0","1","2": 1 1 1 1 1 1 1 1 1 1 ...
```

Looking at the output shows that some factor levels of some predictors are not informative i.e. temp. Also, trying to show all the frequency distributions means making a table with 2^31 elements (2147483648). As a result going to focus on the frequency distribution of temp, date, and precip. Change the integer to actual values using the record from car pacakage.

```
soybean2 <- Soybean
table(soybean2$temp, useNA = "always")
```

```
##
##    0    1    2 <NA>
##   80  374  199   30
```

```
library(car)
soybean2$temp <- recode(soybean2$temp,
                        "0 = 'low'; 1 = 'norm'; 2 = 'high'; NA = 'missing'",
                        levels = c("low", "norm", "high", "missing"))
table(soybean2$temp)
```

```
##
##     low    norm    high missing
##      80     374     199      30
```

```
table(soybean2$date, useNA = "always")
```

```
##
##    0    1    2    3    4    5    6 <NA>
##   26   75   93  118  131  149   90    1
```

```
soybean2$date <- recode(soybean2$date,
                        "0 ='apr';1='may';2='june';3='july';4='aug';5='sept';6='oct';NA = 'missing'",
                        levels = c("apr", "may", "june", "july", "aug", "sept", "missing"))
table(soybean2$date)
```

```
##
##     apr     may    june    july     aug    sept missing
##      26      75      93     118     131     149       1
```

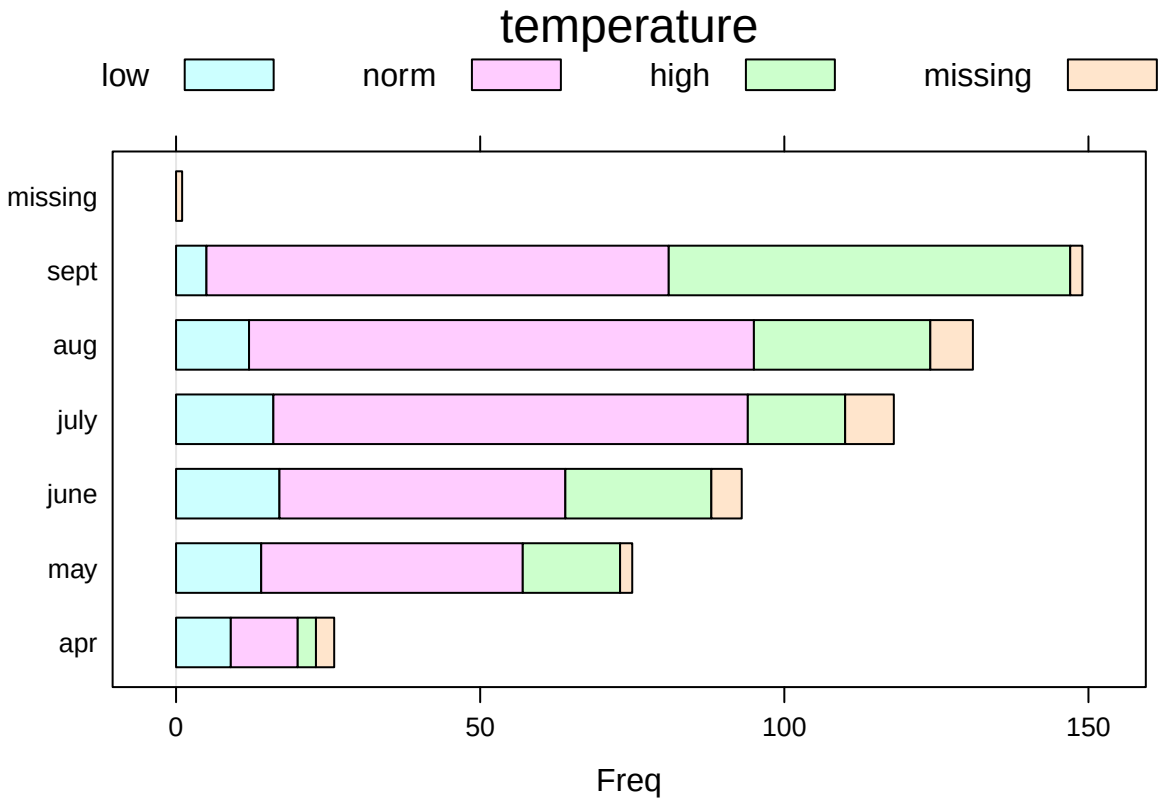
```
table(soybean2$precip, useNA = "always")
```

```
##
##    0    1    2 <NA>
##   74  112  459   38
```

```
soybean2$precip <- recode(soybean2$precip,
                        "0 = 'low'; 1 = 'norm'; 2 = 'high'; NA = 'missing'",
                        levels = c("low", "norm", "high", "missing"))
table(soybean2$precip)
```

```
##
##     low    norm    high missing
##      74     112     459      38
```

```
barchart(table(soybean2$date, soybean2$temp),auto.key = list(columns = 4, title = "temperature"))
```



Frequency

```
table(Soybean$Class, complete.cases(Soybean))
```

```
##
##                               FALSE TRUE
##   2-4-d-injury                   16    0
##   alternarialeaf-spot             0   91
##   anthracnose                     0   44
##   bacterial-blight                0   20
##   bacterial-pustule               0   20
##   brown-spot                      0   92
##   brown-stem-rot                  0   44
##   charcoal-rot                    0   20
##   cyst-nematode                  14    0
##   diaporthe-pod-&-stem-blight    15    0
##   diaporthe-stem-canker           0   20
##   downy-mildew                    0   20
##   frog-eye-leaf-spot              0   91
##   herbicide-injury                8    0
##   phyllosticta-leaf-spot          0   20
##   phytophthora-rot               68   20
##   powdery-mildew                  0   20
##   purple-seed-stain               0   20
##   rhizoctonia-root-rot            0   20
```

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?

First take a look at the missing data distribution. The results show that some classes are more problematic than others:

```
hasMissing <- unlist(lapply(Soybean, function(x) any(is.na(x))))
hasMissing <- names(hasMissing)[hasMissing]
head(hasMissing)
```

```
## [1] "date"        "plant.stand" "precip"      "temp"        "hail"
## [6] "crop.hist"
```

There are several classes where all of the samples have at least one missing predictor value. It is important to know if they are in a single predictor that can be removed or multiple. First get the percentage of missing values for each predictor by class:

```
predclass <- apply(Soybean[, hasMissing], 2, function(x, y)
  {
  tab <- table(is.na(x), y)
  tab[2,]/apply(tab, 2, sum)
  }, y = Soybean$Class)

## eliminate any rows and columns with no missing values

predclass <- predclass[apply(predclass, 1, sum) > 0,]
predclass <- predclass[, apply(predclass, 2, sum) > 0]

t(predclass)
```

```
##                 2-4-d-injury cyst-nematode diaporthe-pod-&-stem-blight
## date                  0.0625             0                         0.0
## plant.stand           1.0000             1                         0.4
## precip                1.0000             1                         0.0
## temp                  1.0000             1                         0.0
## hail                  1.0000             1                         1.0
## crop.hist             1.0000             0                         0.0
## area.dam              0.0625             0                         0.0
## sever                 1.0000             1                         1.0
## seed.tmt              1.0000             1                         1.0
## germ                  1.0000             1                         0.4
## plant.growth          1.0000             0                         0.0
## leaf.halo             0.0000             1                         1.0
## leaf.marg             0.0000             1                         1.0
## leaf.size             0.0000             1                         1.0
## leaf.shread           1.0000             1                         1.0
## leaf.malf             0.0000             1                         1.0
## leaf.mild             1.0000             1                         1.0
## stem                  1.0000             0                         0.0
## lodging               1.0000             1                         1.0
## stem.cankers          1.0000             1                         0.0
## canker.lesion         1.0000             1                         0.0
## fruiting.bodies       1.0000             1                         0.0
## ext.decay             1.0000             1                         0.0
## mycelium              1.0000             1                         0.0
## int.discolor          1.0000             1                         0.0
## sclerotia             1.0000             1                         0.0
## fruit.pods            1.0000             0                         0.0
## fruit.spots           1.0000             1                         0.0
## seed                  1.0000             0                         0.0
## mold.growth           1.0000             0                         0.0
## seed.discolor         1.0000             1                         0.0
## seed.size             1.0000             0                         0.0
## shriveling            1.0000             1                         0.0
## roots                 1.0000             0                         1.0
##                 herbicide-injury phytophthora-rot
## date                           0        0.0000000
## plant.stand                    0        0.0000000
## precip                         1        0.0000000
## temp                           0        0.0000000
## hail                           1        0.7727273
## crop.hist                      0        0.0000000
## area.dam                       0        0.0000000
## sever                          1        0.7727273
## seed.tmt                       1        0.7727273
## germ                           1        0.7727273
## plant.growth                   0        0.0000000
## leaf.halo                      0        0.6250000
## leaf.marg                      0        0.6250000
## leaf.size                      0        0.6250000
## leaf.shread                    0        0.6250000
## leaf.malf                      0        0.6250000
## leaf.mild                      1        0.6250000
## stem                           0        0.0000000
```

```
## lodging                      1        0.7727273
## stem.cankers                 1        0.0000000
## canker.lesion                1        0.0000000
## fruiting.bodies              1        0.7727273
## ext.decay                    1        0.0000000
## mycelium                     1        0.0000000
## int.discolor                 1        0.0000000
## sclerotia                    1        0.0000000
## fruit.pods                   0        0.7727273
## fruit.spots                  1        0.7727273
## seed                         1        0.7727273
## mold.growth                  1        0.7727273
## seed.discolor                1        0.7727273
## seed.size                    1        0.7727273
## shriveling                   1        0.7727273
## roots                        0        0.0000000
```

There are many predictors completely missing for the, 2-4-d-injury, cyst-nematode, and herbicide-injury classes. The phytophthora-rot class has a high rate of missing data across many predictors and the diaporthe-pod-&-stem-blight has a more moderate pattern of missing data.

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

One way to handling missing data is to use an imputation technique. Imputation will not help since almost 100 percent of the predictor values will need to be imputed in a few cases. Another way is to subset the missing as another level or remove the classes associated with the high rate of missing values from the data.

KJ 4.4 Brodnjak-Vonina et al. develop a methodology for food laboratories to determine the type of oil from a sample. Load in the data

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.1.3
```

```
## Loading required package: ggplot2
```

```
data(oil)
str(oilType)
```

```
##  Factor w/ 7 levels "A","B","C","D",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
table(oilType)
```

```
## oilType
##  A  B  C  D  E  F  G
## 37 26  3  7 11 10  2
```

a) Use the sample function in base R create a completely random sample of 60 oils and look how closely the frequencies of the random sample match of the original samples. Repeat to understand variation in the sampling process.

Create 20 splits:

```
set.seed(629)
oilSplits <- vector(mode = "list", length = 20)
for(i in seq(along = oilSplits)) oilSplits[[i]] <- table(sample(oilType, size = 60))
head(oilSplits, 3)
```

```
## [[1]]
##
##  A  B  C  D  E  F  G
## 25 16  3  4  7  4  1
##
## [[2]]
##
##  A  B  C  D  E  F  G
## 24 15  0  6 10  4  1
##
## [[3]]
##
##  A  B  C  D  E  F  G
## 22 17  1  4  8  7  1
```

```
##combine
oilSplits <- do.call("rbind", oilSplits)
head(oilSplits, 3)
```

```
##       A  B C D  E F G
## [1,] 25 16 3 4  7 4 1
## [2,] 24 15 0 6 10 4 1
## [3,] 22 17 1 4  8 7 1
```

```
## distribution of classes
summary(oilSplits/60)
```

```
##        A                B                C                 D
##  Min.   :0.3167   Min.   :0.2333   Min.   :0.00000   Min.   :0.05000
##  1st Qu.:0.3625   1st Qu.:0.2500   1st Qu.:0.01667   1st Qu.:0.06667
##  Median :0.3750   Median :0.2667   Median :0.03333   Median :0.07500
##  Mean   :0.3850   Mean   :0.2708   Mean   :0.03000   Mean   :0.07583
##  3rd Qu.:0.4042   3rd Qu.:0.2833   3rd Qu.:0.05000   3rd Qu.:0.08333
##  Max.   :0.4833   Max.   :0.3500   Max.   :0.05000   Max.   :0.10000
##        E                 F                 G
##  Min.   :0.08333   Min.   :0.06667   Min.   :0.00000
##  1st Qu.:0.10000   1st Qu.:0.07917   1st Qu.:0.01667
##  Median :0.11667   Median :0.10000   Median :0.01667
##  Mean   :0.12000   Mean   :0.10000   Mean   :0.01833
##  3rd Qu.:0.13333   3rd Qu.:0.11667   3rd Qu.:0.03333
##  Max.   :0.16667   Max.   :0.15000   Max.   :0.03333
```

Using a stratified random sample using createDataPartition:

```
set.seed(629)
oilSplits2 <- createDataPartition(oilType, p = .60, times = 20)
oilSplits2 <- lapply(oilSplits2, function(x, y) table(y[x]), y = oilType)
head(oilSplits2, 3)
```

```
## $Resample01
##
##  A  B  C  D  E  F  G
## 23 16  2  5  7  6  2
##
## $Resample02
##
##  A  B  C  D  E  F  G
## 23 16  2  5  7  6  2
##
## $Resample03
##
##  A  B  C  D  E  F  G
## 23 16  2  5  7  6  2
```

```
oilSplits2 <- do.call("rbind", oilSplits2)
summary(oilSplits2/60)
```

```
##        A                B                C                D
##  Min.   :0.3833   Min.   :0.2667   Min.   :0.03333   Min.   :0.08333
##  1st Qu.:0.3833   1st Qu.:0.2667   1st Qu.:0.03333   1st Qu.:0.08333
##  Median :0.3833   Median :0.2667   Median :0.03333   Median :0.08333
##  Mean   :0.3833   Mean   :0.2667   Mean   :0.03333   Mean   :0.08333
##  3rd Qu.:0.3833   3rd Qu.:0.2667   3rd Qu.:0.03333   3rd Qu.:0.08333
##  Max.   :0.3833   Max.   :0.2667   Max.   :0.03333   Max.   :0.08333
##        E                F              G
##  Min.   :0.1167   Min.   :0.1   Min.   :0.03333
##  1st Qu.:0.1167   1st Qu.:0.1   1st Qu.:0.03333
##  Median :0.1167   Median :0.1   Median :0.03333
##  Mean   :0.1167   Mean   :0.1   Mean   :0.03333
##  3rd Qu.:0.1167   3rd Qu.:0.1   3rd Qu.:0.03333
##  Max.   :0.1167   Max.   :0.1   Max.   :0.03333
```

The sampling done using createDataPartition has much less variability that using the sample function. Each partition has at least one sample in each class.

c) What are the options for determining performance of the model?

For determining performance of the model one way would be leave–one–out cross–validation, with the exception of class G, each class is represented in each resample. Some classification models require at least one sample from each class, so resampling these data may place a restriction one which models can be used.

For a test set, leave–one–out cross–validation is a method for assessing performance. A test set could be used if it only consisted of the classes with the most samples although this would only protect against gross overfitting.

d) Look at different sample sizes and accuracy rates to understand the trade-off bteween the uncertainty in the results, the model performance, and the test set size.

```r
binom.test(16,20)
```

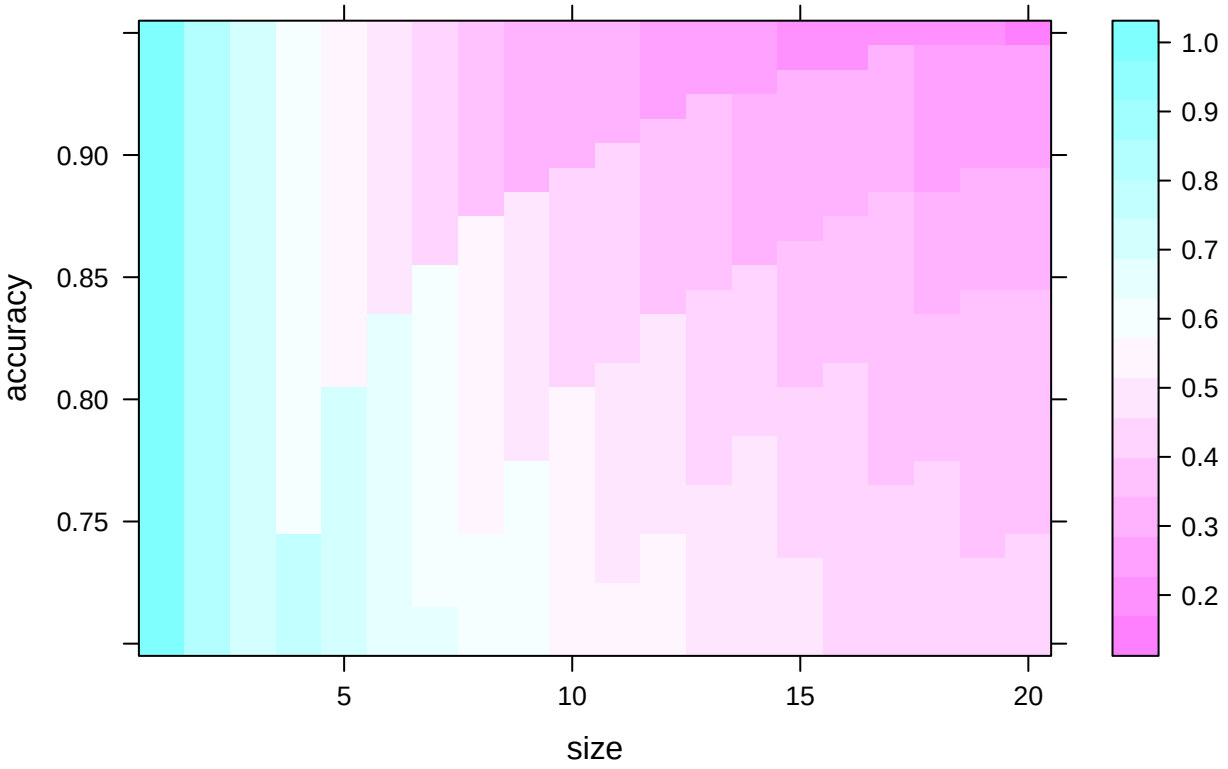
```
##
##  Exact binomial test
##
## data:  16 and 20
## number of successes = 16, number of trials = 20, p-value = 0.01182
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.563386 0.942666
## sample estimates:
## probability of success
##                    0.8
```

Below is the code for the width plot of a binomial confidence interval for overall accuracy for different sample sizes and accuracy rates.

```r
getWidth <- function(values) {
  binom.test(x = floor(values["size"]*values["accuracy"])+1,
             n = values["size"])$conf.int
  }

cigraph <- expand.grid(size = 1:20, accuracy = seq(.7, .95, by = 0.01))
ciWidths <- t(apply(cigraph, 1, getWidth))
head(ciWidths)
```

```
##           [,1]      [,2]
## [1,] 0.0250000 1.0000000
## [2,] 0.1581139 1.0000000
## [3,] 0.2924018 1.0000000
## [4,] 0.1941204 0.9936905
## [5,] 0.2835821 0.9949492
## [6,] 0.3587654 0.9957893
```

```r
cigraph$length <- ciWidths[,2] - ciWidths[,1]
levelplot(length ~ size * accuracy, data = cigraph)
```



Notes: This article was used for understain the trade-off between the uncertainity in the results, model performance and test set size. paper here