Homework 10

#Question 14.1

The breast cancer data set breast-cancer-wisconsin.data.txt from http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/ (description at http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29) has missing values. 1 Use the mean/mode imputation method to impute values for the missing data. 2 Use regression to impute values for the missing data. 3 Use regression with perturbation to impute values for the missing data. 4 (Optional) Compare the results and quality of classification models (e.g., SVM, KNN) build using (1) the data sets from questions 1,2,3; (2) the data that remains after data points with missing values are removed; and (3) the data set when a binary variable is introduced to indicate missing values.

So I'm choosing to do Steps 1 2 and 3 as well as steps 4.1 as 4 is optional, but this is rather short. The good news is Mice will end up doing a majority of the heavy lifting for this analysis. For the time being lets import the data, and assign the column names and for my own santiy 2 is benign 4 in malignant.

```
library(tree)
library(mice)
```

```
## Warning: package 'mice' was built under R version 3.6.3
```

```
##
## Attaching package: 'mice'
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

```
Data = read.csv("breast-cancer-wisconsin.data.txt",header = FALSE,na.strings = "?")
head(Data, n = 5)
```

```
##        V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11
## 1 1000025  5  1  1  1  2  1  3  1   1   2
## 2 1002945  5  4  4  5  7 10  3  2   1   2
## 3 1015425  3  1  1  1  2  2  3  1   1   2
## 4 1016277  6  8  8  1  3  4  3  7   1   2
## 5 1017023  4  1  1  3  2  1  3  1   1   2
```

```
summary(Data)
```

```
##        V1                 V2               V3               V4
##  Min.   :   61634   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000
##  1st Qu.:  870688   1st Qu.: 2.000   1st Qu.: 1.000   1st Qu.: 1.000
##  Median : 1171710   Median : 4.000   Median : 1.000   Median : 1.000
##  Mean   : 1071704   Mean   : 4.418   Mean   : 3.134   Mean   : 3.207
##  3rd Qu.: 1238298   3rd Qu.: 6.000   3rd Qu.: 5.000   3rd Qu.: 5.000
##  Max.   :13454352   Max.   :10.000   Max.   :10.000   Max.   :10.000
##
##        V5               V6               V7               V8
##  Min.   : 1.000   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000
##  1st Qu.: 1.000   1st Qu.: 2.000   1st Qu.: 1.000   1st Qu.: 2.000
##  Median : 1.000   Median : 2.000   Median : 1.000   Median : 3.000
```

```
##  Mean   : 2.807   Mean   : 3.216   Mean   : 3.545   Mean   : 3.438
##  3rd Qu.: 4.000   3rd Qu.: 4.000   3rd Qu.: 6.000   3rd Qu.: 5.000
##  Max.   :10.000   Max.   :10.000   Max.   :10.000   Max.   :10.000
##                                    NA's   :16
##        V9              V10             V11
##  Min.   : 1.000   Min.   : 1.000   Min.   :2.00
##  1st Qu.: 1.000   1st Qu.: 1.000   1st Qu.:2.00
##  Median : 1.000   Median : 1.000   Median :2.00
##  Mean   : 2.867   Mean   : 1.589   Mean   :2.69
##  3rd Qu.: 4.000   3rd Qu.: 1.000   3rd Qu.:4.00
##  Max.   :10.000   Max.   :10.000   Max.   :4.00
##
```

```
colnames(Data) = c("ID Num",
                   "Clump Thickness",
                   "Uniformity of Cell Size",
                   "Uniformity of Cell Shape",
                "Marginal Adhesion",
                "Single Epithelial Cell Size",
                "Bare_Nuclei",
                "Bland Chromatin",
                "Normal Nucleoli",
                "Mitoses",
                "Class")


summary(Data)
```

```
##      ID Num         Clump Thickness  Uniformity of Cell Size
##  Min.   :   61634   Min.   : 1.000   Min.   : 1.000
##  1st Qu.:  870688   1st Qu.: 2.000   1st Qu.: 1.000
##  Median : 1171710   Median : 4.000   Median : 1.000
##  Mean   : 1071704   Mean   : 4.418   Mean   : 3.134
##  3rd Qu.: 1238298   3rd Qu.: 6.000   3rd Qu.: 5.000
##  Max.   :13454352   Max.   :10.000   Max.   :10.000
##
##  Uniformity of Cell Shape Marginal Adhesion Single Epithelial Cell Size
##  Min.   : 1.000           Min.   : 1.000    Min.   : 1.000
##  1st Qu.: 1.000           1st Qu.: 1.000    1st Qu.: 2.000
##  Median : 1.000           Median : 1.000    Median : 2.000
##  Mean   : 3.207           Mean   : 2.807    Mean   : 3.216
##  3rd Qu.: 5.000           3rd Qu.: 4.000    3rd Qu.: 4.000
##  Max.   :10.000           Max.   :10.000    Max.   :10.000
##
##   Bare_Nuclei     Bland Chromatin  Normal Nucleoli     Mitoses
##  Min.   : 1.000   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000
##  1st Qu.: 1.000   1st Qu.: 2.000   1st Qu.: 1.000   1st Qu.: 1.000
##  Median : 1.000   Median : 3.000   Median : 1.000   Median : 1.000
##  Mean   : 3.545   Mean   : 3.438   Mean   : 2.867   Mean   : 1.589
##  3rd Qu.: 6.000   3rd Qu.: 5.000   3rd Qu.: 4.000   3rd Qu.: 1.000
##  Max.   :10.000   Max.   :10.000   Max.   :10.000   Max.   :10.000
##  NA's   :16
##      Class
##  Min.   :2.00
##  1st Qu.:2.00
```

```
##  Median :2.00
##  Mean   :2.69
##  3rd Qu.:4.00
##  Max.   :4.00
##
```

```
Total = nrow(Data)

16/Total
```

```
## [1] 0.02288984
```

So now we're going to get in the weeds.So now for the this test we're going to use M(Imputations) = 3 as the rule of thumb is 3-10 I believe historically some people did M = 100, but that's before my time so I'm a little unsure. Something I will note is this is not necessarily the most accurate way to calculate this.There's actually a relatively "newer" paper that discusses this topic https://arxiv.org/abs/1608.05406 and normally i would use the formula listed here, but since this formula requires us to make forecasts about how much the data would change on new collections of the data I'm not informed enough to make that estimate. Realistically Based on the summary above we're looking at less than 2% of the data missing and the only way I can think of to make this insight is to guess in the dark, so I think it's best to stick with rule of thumb here.

For this we're going to need to re read the data since the column names from above will cause issues. So Lets do that the and then run through the 3 forms of imputation.

```
Data = read.csv("breast-cancer-wisconsin.data.txt",header = FALSE,na.strings = "?")
head(Data, n = 5)
```

```
##        V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11
## 1 1000025  5  1  1  1  2  1  3  1   1   2
## 2 1002945  5  4  4  5  7 10  3  2   1   2
## 3 1015425  3  1  1  1  2  2  3  1   1   2
## 4 1016277  6  8  8  1  3  4  3  7   1   2
## 5 1017023  4  1  1  3  2  1  3  1   1   2
```

```
Mean = mice(Data, method = 'mean', m = 3)
```

```
##
##  iter imp variable
##   1   1  V7
##   1   2  V7
##   1   3  V7
##   2   1  V7
##   2   2  V7
##   2   3  V7
##   3   1  V7
##   3   2  V7
##   3   3  V7
##   4   1  V7
##   4   2  V7
##   4   3  V7
##   5   1  V7
##   5   2  V7
##   5   3  V7
```

```
Regression = mice(Data, method = 'norm.predict', m = 3)
```

```
##
##  iter imp variable
```

```
##   1   1  V7
##   1   2  V7
##   1   3  V7
##   2   1  V7
##   2   2  V7
##   2   3  V7
##   3   1  V7
##   3   2  V7
##   3   3  V7
##   4   1  V7
##   4   2  V7
##   4   3  V7
##   5   1  V7
##   5   2  V7
##   5   3  V7
```

```
Perturbation = mice(Data, method = 'norm.nob', m = 3)
```

```
##
##  iter imp variable
##   1   1  V7
##   1   2  V7
##   1   3  V7
##   2   1  V7
##   2   2  V7
##   2   3  V7
##   3   1  V7
##   3   2  V7
##   3   3  V7
##   4   1  V7
##   4   2  V7
##   4   3  V7
##   5   1  V7
##   5   2  V7
##   5   3  V7
```

```
Mean$imp
```

```
## $V1
## [1] 1 2 3
## <0 rows> (or 0-length row.names)
##
## $V2
## [1] 1 2 3
## <0 rows> (or 0-length row.names)
##
## $V3
## [1] 1 2 3
## <0 rows> (or 0-length row.names)
##
## $V4
## [1] 1 2 3
## <0 rows> (or 0-length row.names)
##
## $V5
```

```
## [1] 1 2 3
## <0 rows> (or 0-length row.names)
##
## $V6
## [1] 1 2 3
## <0 rows> (or 0-length row.names)
##
## $V7
##            1        2        3
## 24  3.544656 3.544656 3.544656
## 41  3.544656 3.544656 3.544656
## 140 3.544656 3.544656 3.544656
## 146 3.544656 3.544656 3.544656
## 159 3.544656 3.544656 3.544656
## 165 3.544656 3.544656 3.544656
## 236 3.544656 3.544656 3.544656
## 250 3.544656 3.544656 3.544656
## 276 3.544656 3.544656 3.544656
## 293 3.544656 3.544656 3.544656
## 295 3.544656 3.544656 3.544656
## 298 3.544656 3.544656 3.544656
## 316 3.544656 3.544656 3.544656
## 322 3.544656 3.544656 3.544656
## 412 3.544656 3.544656 3.544656
## 618 3.544656 3.544656 3.544656
##
## $V8
## [1] 1 2 3
## <0 rows> (or 0-length row.names)
##
## $V9
## [1] 1 2 3
## <0 rows> (or 0-length row.names)
##
## $V10
## [1] 1 2 3
## <0 rows> (or 0-length row.names)
##
## $V11
## [1] 1 2 3
## <0 rows> (or 0-length row.names)
```

```
Regression$imp
```

```
## $V1
## [1] 1 2 3
## <0 rows> (or 0-length row.names)
##
## $V2
## [1] 1 2 3
## <0 rows> (or 0-length row.names)
##
## $V3
## [1] 1 2 3
## <0 rows> (or 0-length row.names)
```

```
##
## $V4
## [1] 1 2 3
## <0 rows> (or 0-length row.names)
##
## $V5
## [1] 1 2 3
## <0 rows> (or 0-length row.names)
##
## $V6
## [1] 1 2 3
## <0 rows> (or 0-length row.names)
##
## $V7
##            1        2        3
## 24  7.191237 7.191237 7.191237
## 41  3.419208 3.419208 3.419208
## 140 1.188951 1.188951 1.188951
## 146 1.579936 1.579936 1.579936
## 159 1.260453 1.260453 1.260453
## 165 1.428797 1.428797 1.428797
## 236 1.943842 1.943842 1.943842
## 250 1.562574 1.562574 1.562574
## 276 1.740990 1.740990 1.740990
## 293 6.432884 6.432884 6.432884
## 295 1.303253 1.303253 1.303253
## 298 1.186205 1.186205 1.186205
## 316 2.083334 2.083334 2.083334
## 322 1.469119 1.469119 1.469119
## 412 1.179806 1.179806 1.179806
## 618 1.059370 1.059370 1.059370
##
## $V8
## [1] 1 2 3
## <0 rows> (or 0-length row.names)
##
## $V9
## [1] 1 2 3
## <0 rows> (or 0-length row.names)
##
## $V10
## [1] 1 2 3
## <0 rows> (or 0-length row.names)
##
## $V11
## [1] 1 2 3
## <0 rows> (or 0-length row.names)
```

```
Perturbation$imp
```

```
## $V1
## [1] 1 2 3
## <0 rows> (or 0-length row.names)
##
## $V2
```

```
## [1] 1 2 3
## <0 rows> (or 0-length row.names)
##
## $V3
## [1] 1 2 3
## <0 rows> (or 0-length row.names)
##
## $V4
## [1] 1 2 3
## <0 rows> (or 0-length row.names)
##
## $V5
## [1] 1 2 3
## <0 rows> (or 0-length row.names)
##
## $V6
## [1] 1 2 3
## <0 rows> (or 0-length row.names)
##
## $V7
##               1           2          3
## 24   6.12769921  6.97624508  5.0750969
## 41   3.16089318  3.27193694  4.2416434
## 140  0.79792992  1.83020809  2.2098439
## 146  1.83982216  0.32294647  1.4621790
## 159  1.60074036 -0.05427722  1.0356921
## 165  0.10955248  1.26438139  0.8612435
## 236  3.63861926  0.18580148  3.3037442
## 250  0.21162831  1.16907152  2.8883580
## 276  2.39671424  0.53591609  0.2758741
## 293  6.85974972  6.14295279  5.7686352
## 295 -0.03523539  1.85471785  3.0766049
## 298  4.94978583  2.41163328 -1.0929077
## 316  3.58261406  0.37739151  2.3271479
## 322  2.34778314 -0.78165225  0.8220651
## 412 -0.85606855  0.55815354 -0.7839702
## 618  1.09826443  1.60955556  1.6388015
##
## $V8
## [1] 1 2 3
## <0 rows> (or 0-length row.names)
##
## $V9
## [1] 1 2 3
## <0 rows> (or 0-length row.names)
##
## $V10
## [1] 1 2 3
## <0 rows> (or 0-length row.names)
##
## $V11
## [1] 1 2 3
## <0 rows> (or 0-length row.names)
```

So at this point we have 1,2,3 completed the next steps complete function which will take our implicit imputations and turn them to explicits. In short we're committing. After that we'll run this through a tree model just to have it ran through some classification model and can make some general inputs about how these imputations worked.

```
set.seed(1)

DataMean= complete(Mean)


#This is a weird problem with tree it doesn't like spaced column names
#or column names that contain numbers we're adding
#a _ to correct for this.

colnames(DataMean) = c("ID_Num",
                       "Clump_Thickness",
                       "Uniformity_of_Cell_Size",
                       "Uniformity_of_Cell_Shape",
                    "Marginal_Adhesion",
                    "Single_Epithelial_Cell_Size",
                    "Bare_Nuclei",
                    "Bland_Chromatin",
                    "Normal_Nucleoli",
                    "Mitoses",
                    "Class")

summary(DataMean)
```

```
##      ID_Num         Clump_Thickness  Uniformity_of_Cell_Size
##  Min.   :   61634   Min.   : 1.000   Min.   : 1.000
##  1st Qu.:  870688   1st Qu.: 2.000   1st Qu.: 1.000
##  Median : 1171710   Median : 4.000   Median : 1.000
##  Mean   : 1071704   Mean   : 4.418   Mean   : 3.134
##  3rd Qu.: 1238298   3rd Qu.: 6.000   3rd Qu.: 5.000
##  Max.   :13454352   Max.   :10.000   Max.   :10.000
##  Uniformity_of_Cell_Shape Marginal_Adhesion Single_Epithelial_Cell_Size
##  Min.   : 1.000           Min.   : 1.000    Min.   : 1.000
##  1st Qu.: 1.000           1st Qu.: 1.000    1st Qu.: 2.000
##  Median : 1.000           Median : 1.000    Median : 2.000
##  Mean   : 3.207           Mean   : 2.807    Mean   : 3.216
##  3rd Qu.: 5.000           3rd Qu.: 4.000    3rd Qu.: 4.000
##  Max.   :10.000           Max.   :10.000    Max.   :10.000
##   Bare_Nuclei     Bland_Chromatin  Normal_Nucleoli     Mitoses
##  Min.   : 1.000   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000
##  1st Qu.: 1.000   1st Qu.: 2.000   1st Qu.: 1.000   1st Qu.: 1.000
##  Median : 1.000   Median : 3.000   Median : 1.000   Median : 1.000
##  Mean   : 3.545   Mean   : 3.438   Mean   : 2.867   Mean   : 1.589
##  3rd Qu.: 5.000   3rd Qu.: 5.000   3rd Qu.: 4.000   3rd Qu.: 1.000
##  Max.   :10.000   Max.   :10.000   Max.   :10.000   Max.   :10.000
##      Class
##  Min.   :2.00
##  1st Qu.:2.00
##  Median :2.00
##  Mean   :2.69
##  3rd Qu.:4.00
```

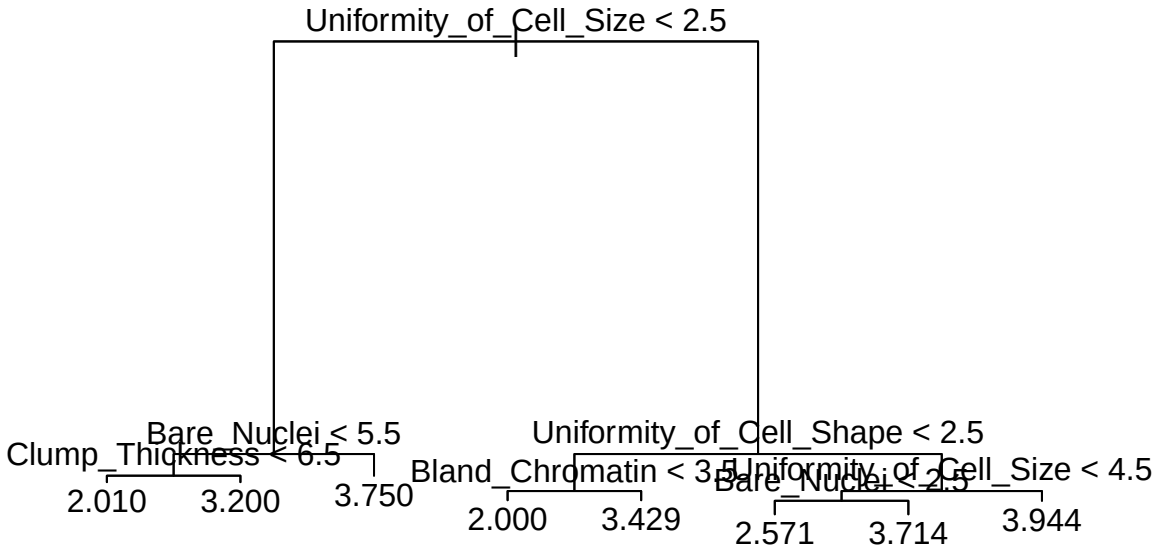
```
##  Max.   :4.00
```

```
tree = tree(Class~., data = DataMean)

summary(tree)
```

```
##
## Regression tree:
## tree(formula = Class ~ ., data = DataMean)
## Variables actually used in tree construction:
## [1] "Uniformity_of_Cell_Size"  "Bare_Nuclei"
## [3] "Clump_Thickness"          "Uniformity_of_Cell_Shape"
## [5] "Bland_Chromatin"
## Number of terminal nodes:  8
## Residual mean deviance:  0.1162 = 80.27 / 691
## Distribution of residuals:
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -1.944000 -0.009615 -0.009615  0.000000  0.056500  1.990000
```

```
plot(tree)
text(tree)
```




```
treepredict = predict(tree, data = DataMean[,1:10])
SSmult = sum((treepredict - DataMean[,11])^2)
SStotal = sum((DataMean[,11] - mean(DataMean[,11]))^2)

rsq1 = 1 - SSmult/SStotal
```

```
DataRegression = complete(Regression)

#This is a weird problem with tree it doesn't like spaced column names
#or column names that contain numbers we're adding
#a _ to correct for this.

colnames(DataRegression) = c("ID_Num",
                     "Clump_Thickness",
                     "Uniformity_of_Cell_Size",
                     "Uniformity_of_Cell_Shape",
                   "Marginal_Adhesion",
                   "Single_Epithelial_Cell_Size",
                   "Bare_Nuclei",
                   "Bland_Chromatin",
                   "Normal_Nucleoli",
                   "Mitoses",
                   "Class")

summary(DataRegression)
```

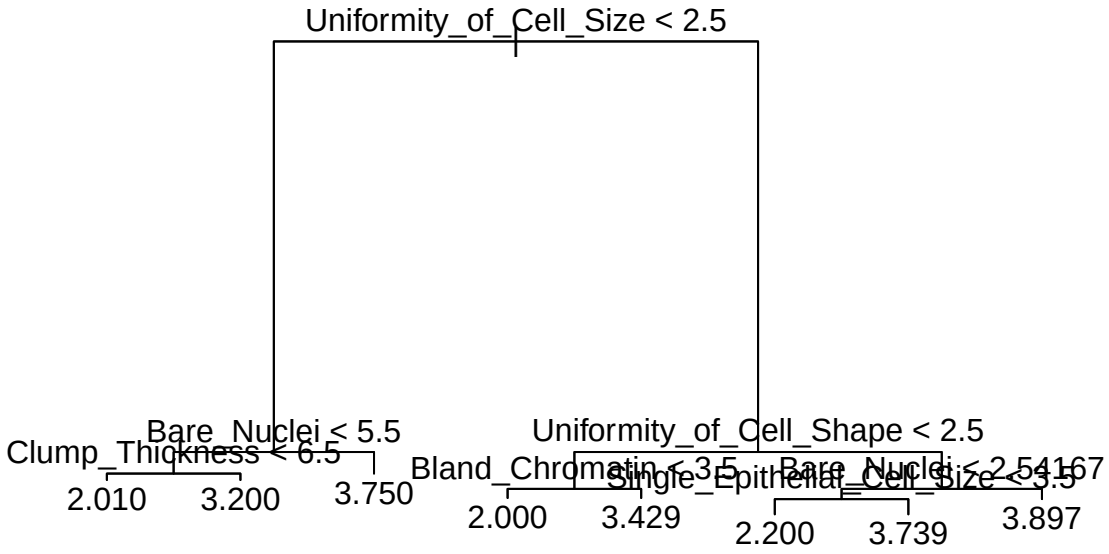
```
##      ID_Num         Clump_Thickness  Uniformity_of_Cell_Size
##  Min.   :   61634   Min.   : 1.000   Min.   : 1.000
##  1st Qu.:  870688   1st Qu.: 2.000   1st Qu.: 1.000
##  Median : 1171710   Median : 4.000   Median : 1.000
##  Mean   : 1071704   Mean   : 4.418   Mean   : 3.134
##  3rd Qu.: 1238298   3rd Qu.: 6.000   3rd Qu.: 5.000
##  Max.   :13454352   Max.   :10.000   Max.   :10.000
##  Uniformity_of_Cell_Shape Marginal_Adhesion Single_Epithelial_Cell_Size
##  Min.   : 1.000           Min.   : 1.000    Min.   : 1.000
##  1st Qu.: 1.000           1st Qu.: 1.000    1st Qu.: 2.000
##  Median : 1.000           Median : 1.000    Median : 2.000
##  Mean   : 3.207           Mean   : 2.807    Mean   : 3.216
##  3rd Qu.: 5.000           3rd Qu.: 4.000    3rd Qu.: 4.000
##  Max.   :10.000           Max.   :10.000    Max.   :10.000
##   Bare_Nuclei     Bland_Chromatin  Normal_Nucleoli     Mitoses
##  Min.   : 1.000   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000
##  1st Qu.: 1.000   1st Qu.: 2.000   1st Qu.: 1.000   1st Qu.: 1.000
##  Median : 1.000   Median : 3.000   Median : 1.000   Median : 1.000
##  Mean   : 3.515   Mean   : 3.438   Mean   : 2.867   Mean   : 1.589
##  3rd Qu.: 6.000   3rd Qu.: 5.000   3rd Qu.: 4.000   3rd Qu.: 1.000
##  Max.   :10.000   Max.   :10.000   Max.   :10.000   Max.   :10.000
##      Class
##  Min.   :2.00
##  1st Qu.:2.00
##  Median :2.00
##  Mean   :2.69
##  3rd Qu.:4.00
##  Max.   :4.00
```

```
tree = tree(Class~., data = DataRegression)

summary(tree)
```

```
##
## Regression tree:
## tree(formula = Class ~ ., data = DataRegression)
## Variables actually used in tree construction:
## [1] "Uniformity_of_Cell_Size"     "Bare_Nuclei"
## [3] "Clump_Thickness"             "Uniformity_of_Cell_Shape"
## [5] "Bland_Chromatin"             "Single_Epithelial_Cell_Size"
## Number of terminal nodes:  8
## Residual mean deviance:  0.1125 = 77.75 / 691
## Distribution of residuals:
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -1.897000 -0.009615 -0.009615  0.000000  0.102800  1.990000
```

```
plot(tree)
text(tree)
```




```
treepredict = predict(tree, data = DataRegression[,1:10])
SSmult = sum((treepredict - DataRegression[,11])^2)
SStotal = sum((DataRegression[,11] - mean(DataRegression[,11]))^2)

SStotal
```

```
## [1] 631.6338
```

```
SSmult
```

```
## [1] 77.74892
```

```
rsq2 = 1 - SSmult/SStotal


DataPerturbation = complete(Perturbation)


#This is a weird problem with tree it doesn't like spaced column names
#or column names that contain numbers we're adding
#a _ to correct for this.

colnames(DataPerturbation) = c("ID_Num",
                    "Clump_Thickness",
                    "Uniformity_of_Cell_Size",
                    "Uniformity_of_Cell_Shape",
                  "Marginal_Adhesion",
                  "Single_Epithelial_Cell_Size",
                  "Bare_Nuclei",
                  "Bland_Chromatin",
                  "Normal_Nucleoli",
                  "Mitoses",
                  "Class")

summary(DataPerturbation)
```

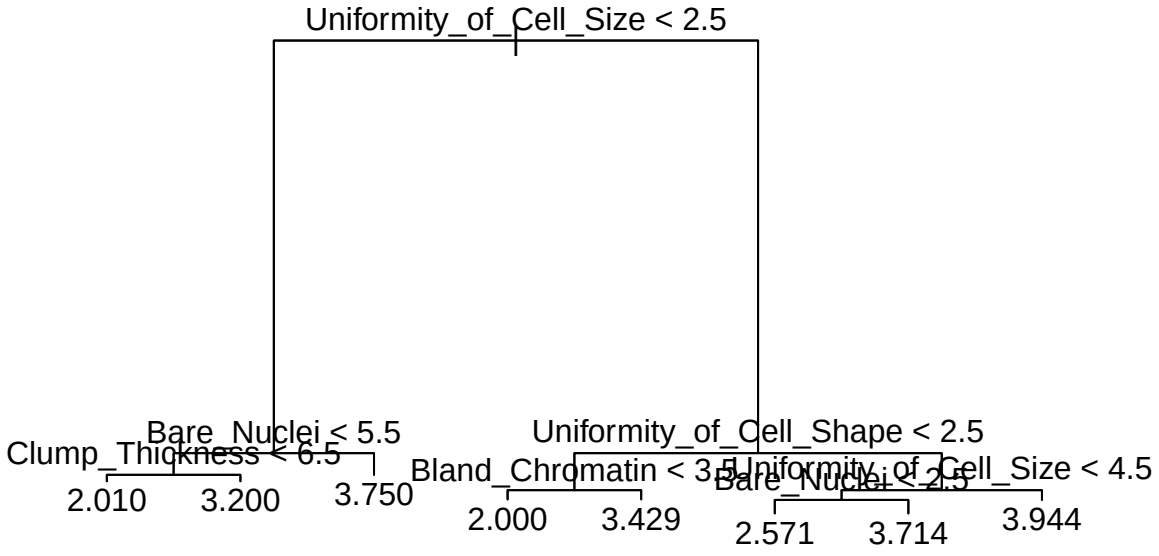
```
##      ID_Num         Clump_Thickness  Uniformity_of_Cell_Size
##  Min.   :   61634   Min.   : 1.000   Min.   : 1.000
##  1st Qu.:  870688   1st Qu.: 2.000   1st Qu.: 1.000
##  Median : 1171710   Median : 4.000   Median : 1.000
##  Mean   : 1071704   Mean   : 4.418   Mean   : 3.134
##  3rd Qu.: 1238298   3rd Qu.: 6.000   3rd Qu.: 5.000
##  Max.   :13454352   Max.   :10.000   Max.   :10.000
##  Uniformity_of_Cell_Shape Marginal_Adhesion Single_Epithelial_Cell_Size
##  Min.   : 1.000           Min.   : 1.000    Min.   : 1.000
##  1st Qu.: 1.000           1st Qu.: 1.000    1st Qu.: 2.000
##  Median : 1.000           Median : 1.000    Median : 2.000
##  Mean   : 3.207           Mean   : 2.807    Mean   : 3.216
##  3rd Qu.: 5.000           3rd Qu.: 4.000    3rd Qu.: 4.000
##  Max.   :10.000           Max.   :10.000    Max.   :10.000
##   Bare_Nuclei      Bland_Chromatin  Normal_Nucleoli     Mitoses
##  Min.   :-0.8561   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000
##  1st Qu.: 1.0000   1st Qu.: 2.000   1st Qu.: 1.000   1st Qu.: 1.000
##  Median : 1.0000   Median : 3.000   Median : 1.000   Median : 1.000
##  Mean   : 3.5176   Mean   : 3.438   Mean   : 2.867   Mean   : 1.589
##  3rd Qu.: 6.0000   3rd Qu.: 5.000   3rd Qu.: 4.000   3rd Qu.: 1.000
##  Max.   :10.0000   Max.   :10.000   Max.   :10.000   Max.   :10.000
##      Class
##  Min.   :2.00
##  1st Qu.:2.00
##  Median :2.00
##  Mean   :2.69
##  3rd Qu.:4.00
##  Max.   :4.00
```

```
tree = tree(Class~., data = DataPerturbation)

summary(tree)
```

```
##
## Regression tree:
## tree(formula = Class ~ ., data = DataPerturbation)
## Variables actually used in tree construction:
## [1] "Uniformity_of_Cell_Size"  "Bare_Nuclei"
## [3] "Clump_Thickness"          "Uniformity_of_Cell_Shape"
## [5] "Bland_Chromatin"
## Number of terminal nodes:  8
## Residual mean deviance:  0.1162 = 80.27 / 691
## Distribution of residuals:
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -1.944000 -0.009615 -0.009615  0.000000  0.056500  1.990000
```

```
plot(tree)
text(tree)
```




```
treepredict = predict(tree, data = DataPerturbation[,1:10])
SSmult = sum((treepredict - DataPerturbation[,11])^2)
SStotal = sum((DataPerturbation[,11] - mean(DataPerturbation[,11]))^2)

SStotal
```

```
## [1] 631.6338
```

```
SSmult
```

```
## [1] 80.268
```

```
rsq3 = 1 - SSmult/SStotal

rsq1
```

```
## [1] 0.87292
```

```
rsq2
```

```
## [1] 0.8769082
```

```
rsq3
```

```
## [1] 0.87292
```

So though these are similar it looks like we can point to one method having a slight advantage but nothing substantial enough to rule out the other methods. We could and should run more models on this as well as the other data imputation methods, but in the interest of keeping this concise we'll stop there. The difference is not enough to rule any other method out so we should run additional models against this, but for the time being we'll end it here.

#Question 15.1

Describe a situation or problem from your job, everyday life, current events, etc., for which optimization would be appropriate. What data would you need?

Optimization modeling in real life is used to shipping in ensure that packages are delivered to a location within a set period to prevent shortages and meet demand. To keep this more focused lets talk about just global timber shipment. So we are looking for the production for each forester group, the demand, the weather conditions, distance, Natural disaster dates, Speed and capacity of boats, capacity of docks in terms of unloading as well as on the ground shipping in country, and I'm sure many many more factors.