Group_Assignment_2.R

skhanna

Mon Nov 05 21:22:32 2018

```
#Group Assignment 2: Algae Blooms



#install.packages("DMwR")

#loading library "DMwR"
library(DMwR)
```

```
## Loading required package: lattice
```

```
## Loading required package: grid
```

```
#Specifying the path .csv file and loading the analysis.data file
algae <- read.csv("C:/Users/SKHANNA/Desktop/Analysis.data",header=F,dec='.',
                  col.names=c('season','size','speed','mxPH','mnO2','Cl','NO3','NH4','oPO4','PO
4','Chla','a1','a2','a3','a4','a5','a6','a7'),
                  na.strings=c('XXXXXXX'))

#views the first few rows of the data for algae
head(algae)
```

```
##   season   size  speed mxPH mnO2     Cl      NO3     NH4    oPO4     PO4
## 1 winter small_ medium 8.00  9.8 60.800  6.23800 578.000 105.000 170.000
## 2 spring small_ medium 8.35  8.0 57.750  1.28800 370.000 428.750 558.750
## 3 autumn small_ medium 8.10 11.4 40.020  5.33000 346.667 125.667 187.057
## 4 spring small_ medium 8.07  4.8 77.364  2.30200  98.182  61.182 138.700
## 5 autumn small_ medium 8.06  9.0 55.350 10.41600 233.700  58.222  97.580
## 6 winter small_ high__ 8.25 13.1 65.750  9.24800 430.000  18.250  56.667
##   Chla   a1   a2   a3  a4   a5   a6  a7
## 1 50.0  0.0  0.0  0.0 0.0 34.2  8.3 0.0
## 2  1.3  1.4  7.6  4.8 1.9  6.7  0.0 2.1
## 3 15.6  3.3 53.6  1.9 0.0  0.0  0.0 9.7
## 4  1.4  3.1 41.0 18.9 0.0  1.4  0.0 1.4
## 5 10.5  9.2  2.9  7.5 0.0  7.5  4.1 1.0
## 6 28.4 15.1 14.6  1.4 0.0 22.5 12.6 2.9
```

```
#loading the eval.data file
eval <- read.csv("C:/Users/SKHANNA/Desktop/eval.data",header=F,dec='.',
                 col.names=c('season','size','speed','mxPH','mnO2','Cl','NO3','NH4','oPO4','PO4'
,'Chla'),
                 na.strings=c('XXXXXXX'))
```

```
## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec =
## dec, : embedded nul(s) found in input
```

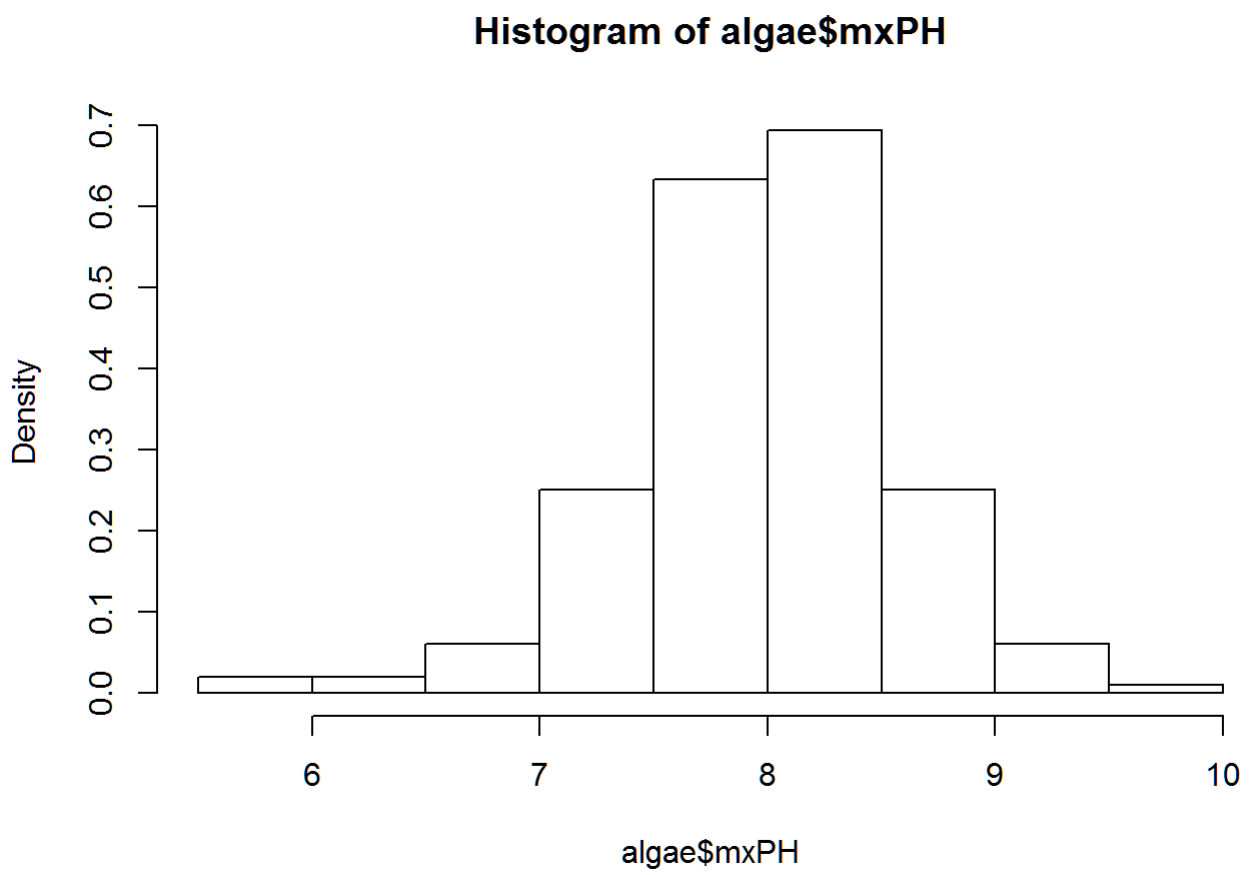
```
#views the first few rows of the data for eval
head(eval)
```

```
##   season   size  speed mxPH mnO2     Cl     NO3     NH4    oPO4     PO4
## 1 summer small_ medium 7.95  5.7 57.333 2.46000 273.333 295.667 380.000
## 2 winter small_ medium 7.98  8.8 59.333 7.39200 286.667  33.333 138.000
## 3 summer small_ medium 8.00  7.2 80.000 1.95700 174.286  47.857 113.714
## 4 spring small_ high__ 8.35  8.4 68.000 3.02600 458.000  45.200 111.800
## 5 spring small_ medium 8.10 13.2 19.000 0.00000 130.000   6.000  40.000
## 6 summer small_ medium 8.37 12.1 12.850 0.84000  15.000   5.000  10.507
##   Chla
## 1   NA
## 2  7.1
## 3  4.5
## 4  3.2
## 5  2.0
## 6 13.8
```

```
#algae <- read.table('Analysis.data',header=F,dec='.',
#                    col.names=c('season','size','speed','mxPH','mnO2','Cl',
#                                'NO3','NH4','oPO4','PO4','Chla','a1','a2','a3','a4',
#                               'a5','a6','a7'),
#                na.strings=c('XXXXXXX'))

#displays the summary of its descriptive analysis
summary(algae)
```

```
##     season       size       speed          mxPH            mnO2
##  autumn:40   large_:45   high__:84   Min.   :5.600   Min.   : 1.500
##  spring:53   medium:84   low___:33   1st Qu.:7.700   1st Qu.: 7.725
##  summer:45   small_:71   medium:83   Median :8.060   Median : 9.800
##  winter:62                           Mean   :8.012   Mean   : 9.118
##                                      3rd Qu.:8.400   3rd Qu.:10.800
##                                      Max.   :9.700   Max.   :13.400
##                                      NA's   :1       NA's   :2
##        Cl               NO3            NH4              oPO4
##  Min.   :  0.222   0.23000:  2   Min.   :  5.00   Min.   :  1.00
##  1st Qu.: 10.981   0.73500:  2   1st Qu.: 35.62   1st Qu.: 16.00
##  Median : 32.730   1.32000:  2   Median : 99.67   Median : 41.40
##  Mean   : 43.636   3.02000:  2   Mean   :154.45   Mean   : 83.33
##  3rd Qu.: 57.824   3.14000:  2   3rd Qu.:203.73   3rd Qu.:102.25
##  Max.   :391.500   (Other):188   Max.   :931.83   Max.   :771.60
##  NA's   :10        NA's   :  2   NA's   :2        NA's   :2
##       PO4              Chla              a1               a2
##  Min.   :  0.90   Min.   :  0.00   Min.   : 0.000   Min.   : 0.000
##  1st Qu.: 19.39   1st Qu.:  2.00   1st Qu.: 1.475   1st Qu.: 0.000
##  Median : 84.50   Median :  5.20   Median : 7.400   Median : 2.100
##  Mean   :111.55   Mean   : 13.54   Mean   :16.863   Mean   : 6.934
##  3rd Qu.:182.16   3rd Qu.: 18.30   3rd Qu.:24.075   3rd Qu.: 9.075
##  Max.   :558.75   Max.   :110.46   Max.   :89.800   Max.   :72.600
##  NA's   :2        NA's   :12
##        a3               a4               a5               a6
##  Min.   : 0.000   Min.   : 0.000   Min.   : 0.00   Min.   : 0.000
##  1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.: 0.00   1st Qu.: 0.000
##  Median : 1.750   Median : 0.000   Median : 1.90   Median : 0.000
##  Mean   : 4.729   Mean   : 1.885   Mean   : 5.63   Mean   : 5.199
##  3rd Qu.: 6.150   3rd Qu.: 2.225   3rd Qu.: 7.70   3rd Qu.: 6.725
##  Max.   :44.600   Max.   :35.600   Max.   :77.60   Max.   :52.500
##
##        a7
##  Min.   : 0.000
##  1st Qu.: 0.000
##  Median : 0.000
##  Mean   : 2.506
##  3rd Qu.: 2.400
##  Max.   :31.600
##  NA's   :17
```

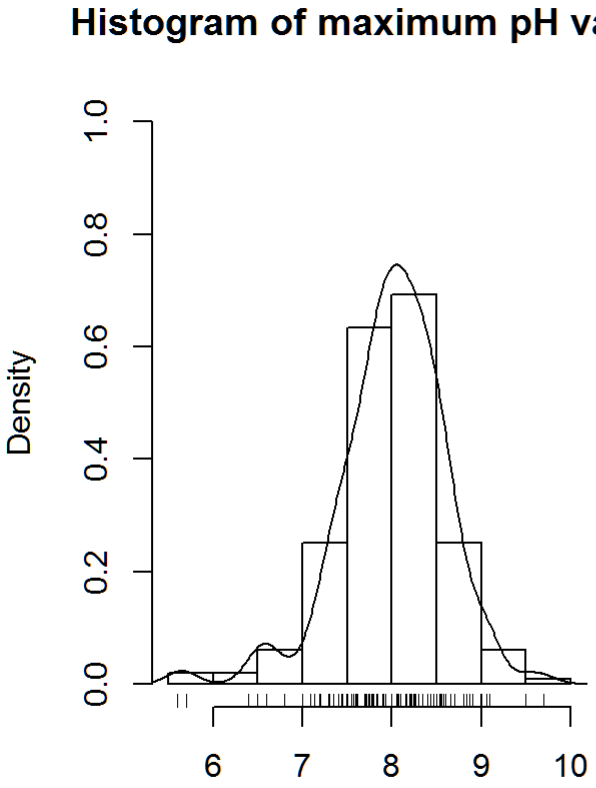
```
#plots the histogram with the variable mxPH and this follows normal distribution
hist(algae$mxPH, prob = T)
```

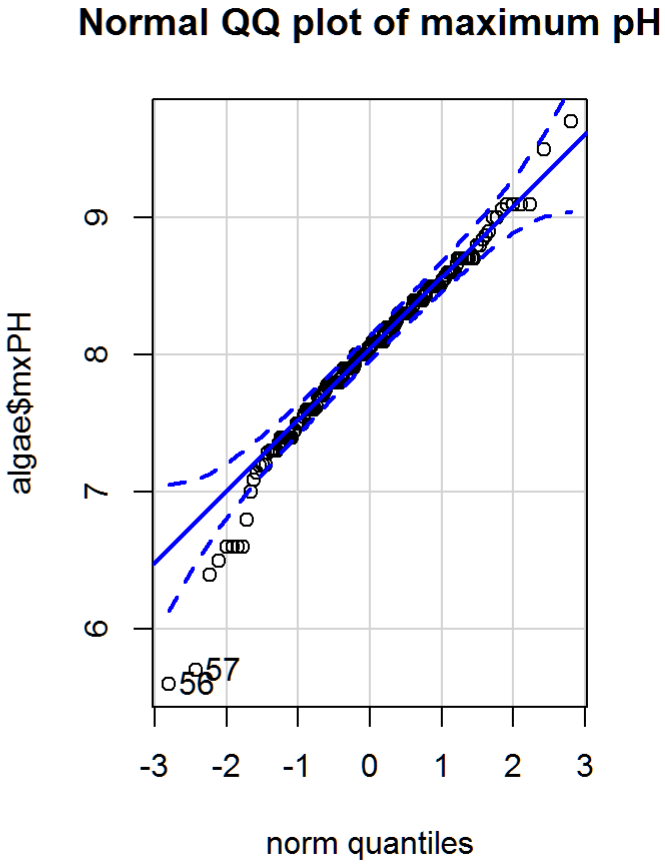



```
#Loads the package 'car'
library(car)
```

```
## Loading required package: carData
```

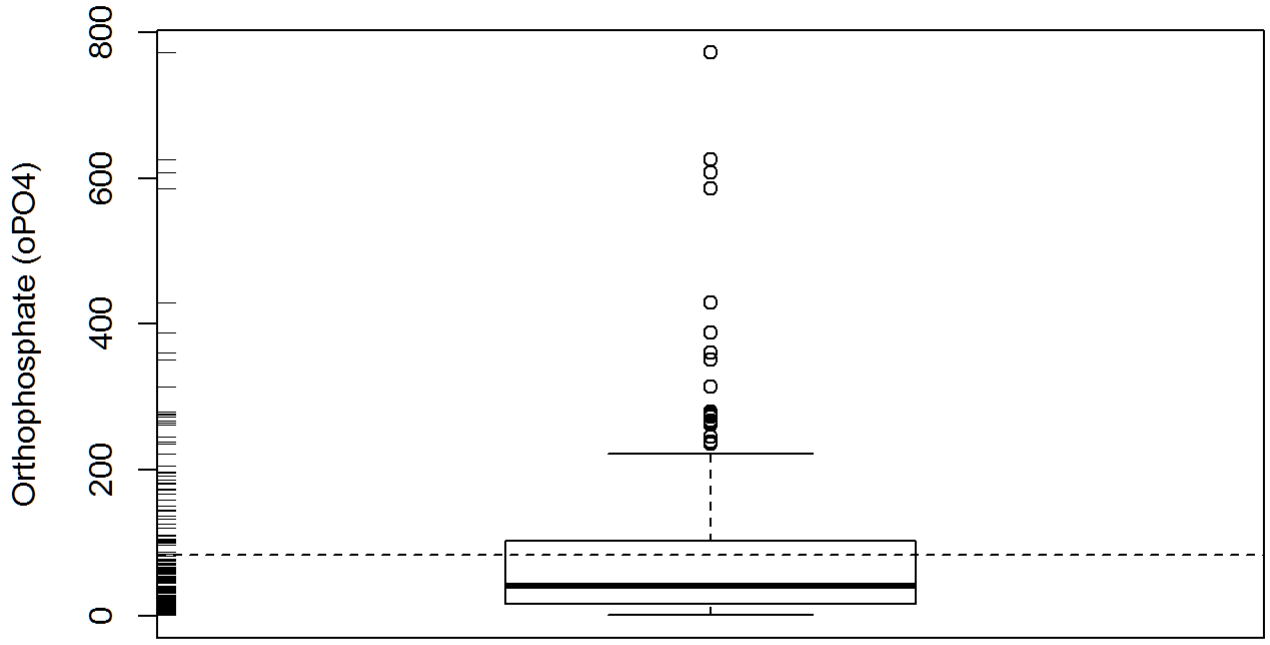
```
#par() is used to set sveral parameters of the R graphics system
par(mfrow=c(1,2))

#plots a histogram but with an empty x-axis label
hist(algae$mxPH, prob=T, xlab='',main='Histogram of maximum pH value',ylim=0:1)
lines(density(algae$mxPH,na.rm=T))
rug(jitter(algae$mxPH))

#plots a smooth version of the histogram
#this allows easy spotting of outliers
qqPlot(algae$mxPH,main='Normal QQ plot of maximum pH')
```



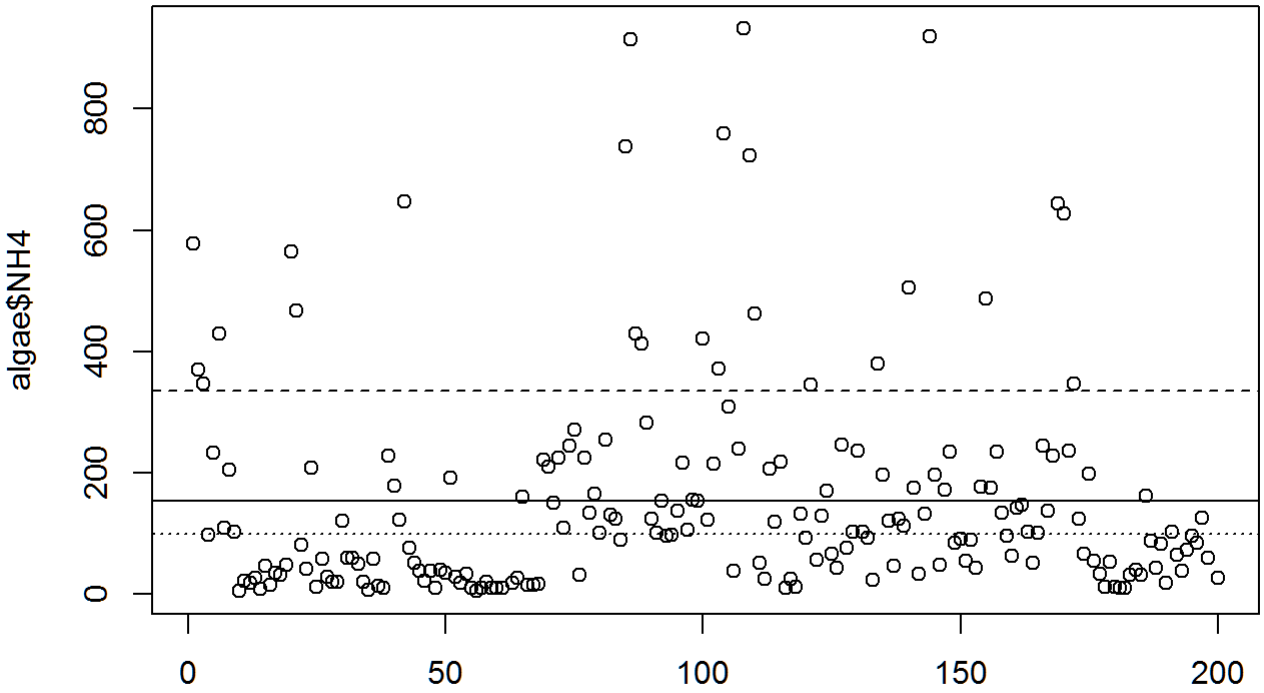



```
## [1] 56 57
```

```
par(mfrow=c(1,1))


#draws a boxplot with variable oPO4
#this provides summary of some key properties of the distribution
boxplot(algae$oPO4, ylab = "Orthophosphate (oPO4)")
rug(jitter(algae$oPO4), side = 2)
abline(h = mean(algae$oPO4, na.rm = T), lty = 2)
```

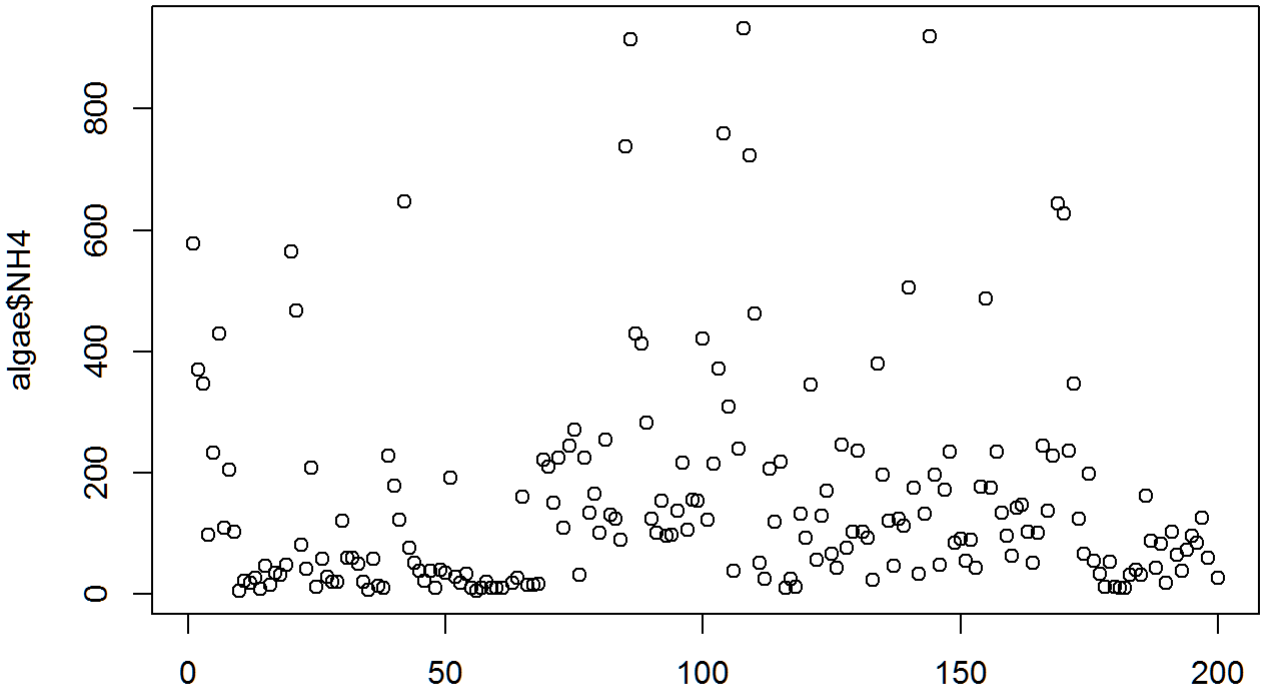



```
#plots all the values of the variables
plot(algae$NH4, xlab = "")
#draws three informative lines
#this is with mean value
abline(h = mean(algae$NH4, na.rm = T), lty = 1)
#mean + one Standard deviation
abline(h = mean(algae$NH4, na.rm = T) + sd(algae$NH4, na.rm = T),
       lty = 2)
#with median
abline(h = median(algae$NH4, na.rm = T), lty = 3)
#writes the respective row number in the algae dataframe
identify(algae$NH4)
```




```
## integer(0)
```

```
# for inspecting the respective observations in the algae data frame
plot(algae$NH4, xlab = "")
clicked.lines <- identify(algae$NH4)
```




```
algae[clicked.lines, ]
```

```
##  [1] season size   speed  mxPH   mnO2   Cl     NO3    NH4    oPO4   PO4
## [11] Chla   a1     a2     a3     a4     a5     a6     a7
## <0 rows> (or 0-length row.names)
```

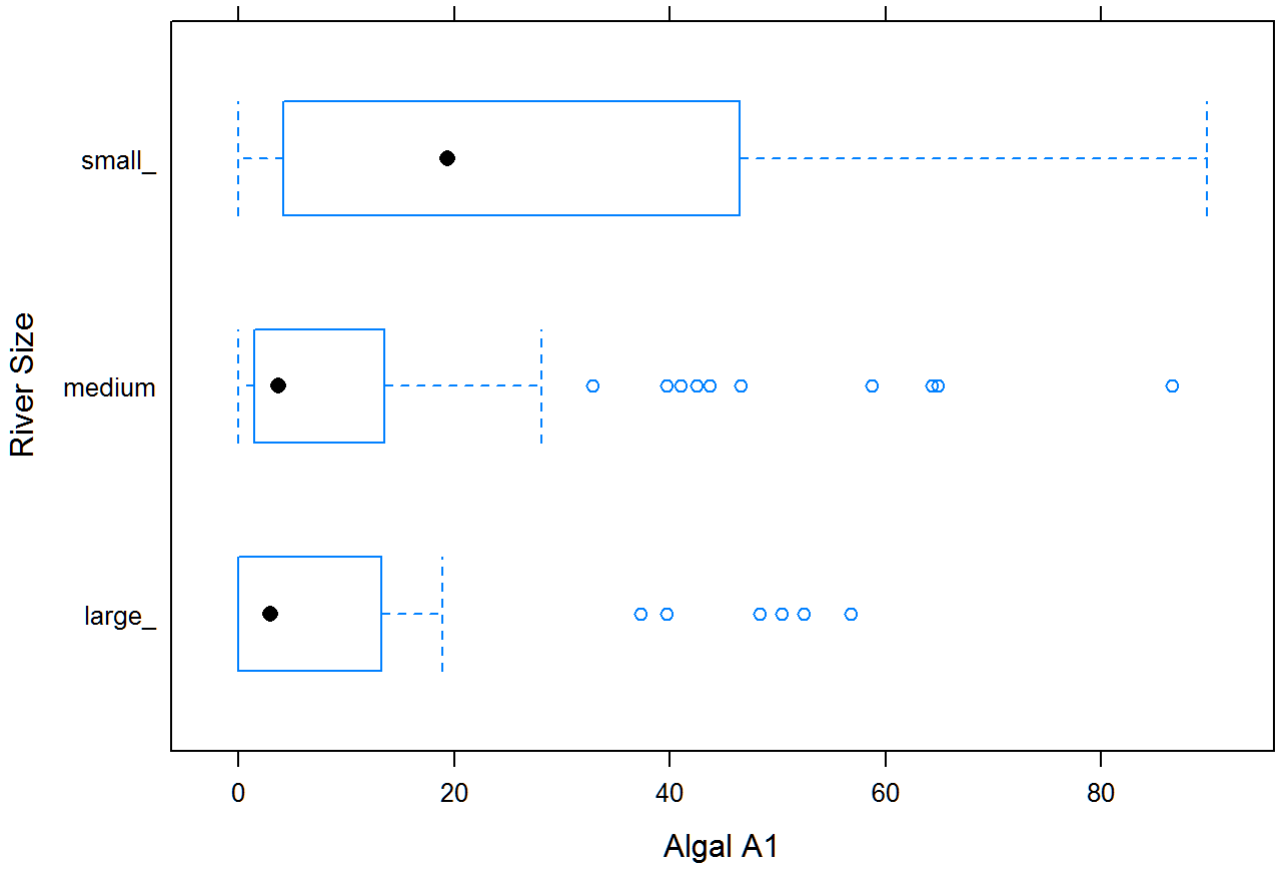
```
#This instruction illustrates another form of indexing a data frame, using a logical expression
 as a row selector
#this gives us the rows of the data frame that have known values in NH4 and are greater than 19,
000.
algae[algae$NH4 > 19000, ]
```

```
##      season size speed mxPH mnO2 Cl  NO3 NH4 oPO4 PO4 Chla a1 a2 a3 a4 a5
## NA     <NA> <NA>  <NA>   NA   NA NA <NA>  NA   NA  NA   NA NA NA NA NA NA
## NA.1   <NA> <NA>  <NA>   NA   NA NA <NA>  NA   NA  NA   NA NA NA NA NA NA
##      a6 a7
## NA   NA NA
## NA.1 NA NA
```

```
algae[!is.na(algae$NH4) & algae$NH4 > 19000,]
```

```
##  [1] season size   speed  mxPH   mnO2   Cl     NO3    NH4    oPO4   PO4
## [11] Chla   a1     a2     a3     a4     a5     a6     a7
## <0 rows> (or 0-length row.names)
```

```
#Loads the "lattice" package
library(lattice)
#we can observe that higher frequencies of algal a1 are expected in smaller rivers
#plots the boxplot
bwplot(size ~ a1, data=algae, ylab='River Size',xlab='Algal A1')
```




```
#install.packages("Hmisc")
#Loads the "Hmisc" package
library(Hmisc)
```

```
## Loading required package: survival
```

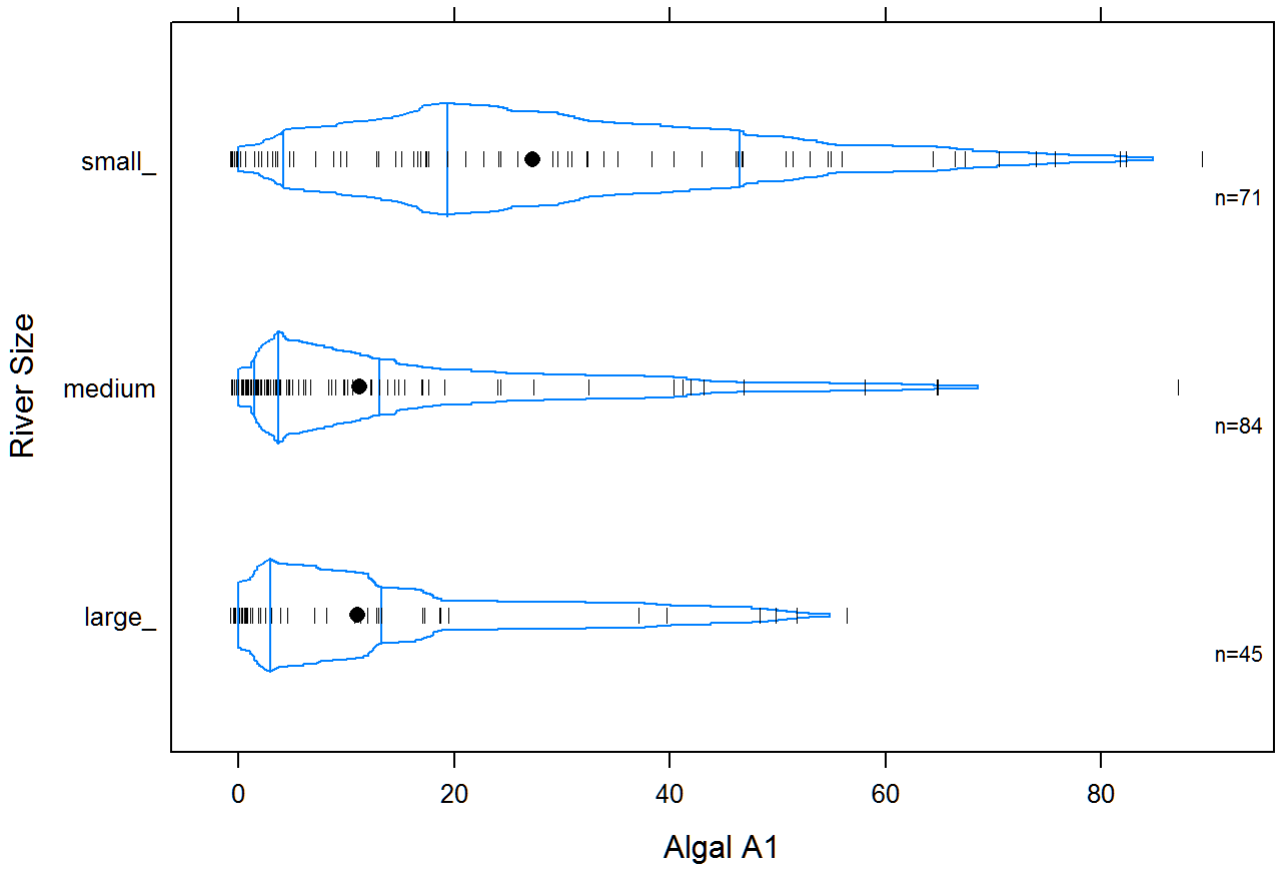
```
## Loading required package: Formula
```

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

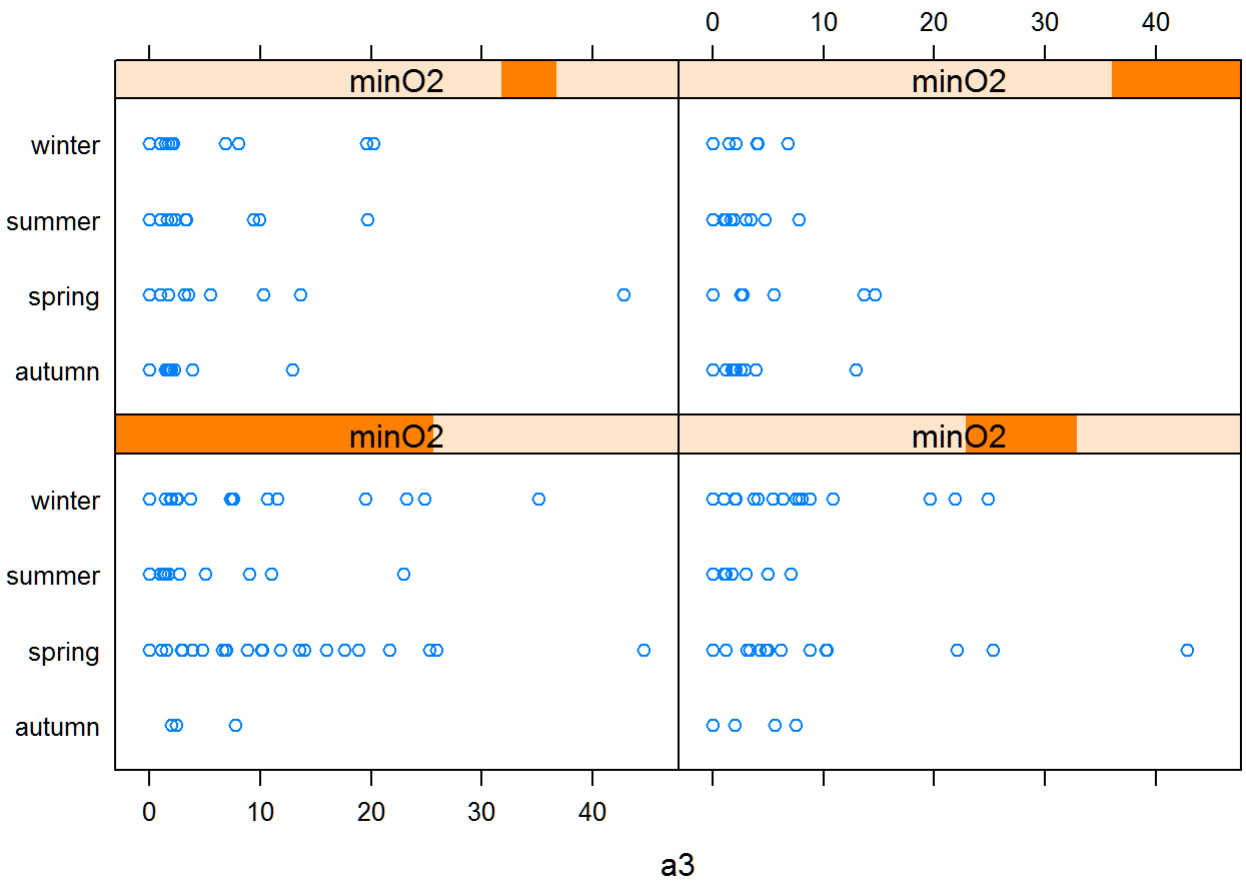
```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
#graph shows us the actual values of data and the information of the distribution of these value
s is provided by the quantile plots
bwplot(size ~ a1, data=algae,panel=panel.bpplot,
       probs=seq(.01,.49,by=.01), datadensity=TRUE,
       ylab='River Size',xlab='Algal A1')
```




```
#plots a conditioned box percentile plot
minO2 <- equal.count(na.omit(algae$mnO2),
                     number=4,overlap=1/5)

#plots a conditioned strip plot using a continuos variable
stripplot(season ~ a3|minO2,
          data=algae[!is.na(algae$mnO2),])
```




```
#The function complete.cases() produces a vector of Boolean values with
#as many elements as there are rows in the algae data frame, where an element
#is true if the respective row is "clean" of NA values
algae[!complete.cases(algae),]
```

```
##     season   size  speed mxPH mnO2      Cl                NO3     NH4
## 20  spring small_ medium 7.79  3.2  64.000   2.822008777.59961 564.600
## 21  winter small_ medium 7.83 10.7  88.000   4.825001729.00000 467.500
## 28  autumn small_ high__ 6.80 11.1   9.000            0.63000  20.000
## 34  autumn small_ medium 8.40  9.9  34.500   2.818003515.00000  20.000
## 35  winter small_ medium 8.27  7.8  29.200   0.050006400.00000   7.400
## 36  summer small_ medium 8.66  8.4  30.523   3.444001911.00000  58.875
## 38  spring small_ high__ 8.00   NA   1.450            0.81000  10.000
## 48  winter small_ low___   NA 12.6   9.000            0.23000  10.000
## 55  winter small_ high__ 6.60 10.8      NA            3.24500  10.000
## 56  spring small_ medium 5.60 11.8      NA            2.22000   5.000
## 57  autumn small_ medium 5.70 10.8      NA            2.55000  10.000
## 58  spring small_ high__ 6.60  9.5      NA            1.32000  20.000
## 59  summer small_ high__ 6.60 10.8      NA            2.64000  10.000
## 60  autumn small_ medium 6.60 11.3      NA            4.17000  10.000
## 61  spring small_ medium 6.50 10.4      NA            5.97000  10.000
## 62  summer small_ medium 6.40   NA      NA               <NA>      NA
## 63  autumn small_ high__ 7.83 11.7   4.083            1.32800  18.000
## 69  winter small_ medium 7.50  1.5  32.400   0.921001386.25000 220.750
## 70  spring small_ medium 7.50  1.8  29.775   1.051002082.85010 209.857
## 71  summer small_ medium 7.80  7.1  32.540   1.720002167.37012 151.125
## 88  winter medium medium 7.80  3.6  48.667   4.030005738.33008 412.333
## 89  summer medium medium 7.60  9.7  53.102   7.160004073.33008 282.167
## 116 winter medium high__ 9.70 10.8   0.222            0.40600  10.000
## 133 winter medium medium 7.90  9.8 194.750   6.513003466.65991  23.000
## 146 autumn medium low___ 7.80  6.5  64.093   7.740001990.16003  47.500
## 153 autumn medium high__ 7.30 11.8  44.205 45.6500024064.00000  44.000
## 156 spring large_ low___ 7.80  3.2  94.000   4.908001131.66003 175.667
## 157 summer large_ low___ 7.60  4.9  69.000   3.685001495.00000 234.500
## 161 spring large_ low___ 9.00  5.8      NA            0.90000 142.000
## 171 winter large_ medium 8.24  6.1  95.367   3.561001168.00000 236.400
## 172 summer large_ medium 7.91  6.2 151.833   3.923001081.66003 346.167
## 184 winter large_ high__ 8.00 10.9   9.055            0.82500  40.000
## 199 winter large_ medium 8.00  7.6      NA               <NA>      NA
##        oPO4     PO4  Chla   a1   a2   a3   a4   a5   a6  a7
## 20  771.600   4.500  0.00  0.0  0.0 44.6  0.0  0.0  1.4  NA
## 21  586.000  16.000  0.00  0.0  0.0  6.8  6.1  0.0  0.0  NA
## 28    4.000      NA  2.70 30.3  1.9  0.0  0.0  2.1  1.4 2.1
## 34   47.000   2.300 13.60  9.1  0.0  0.0  1.4  0.0  0.0  NA
## 35   23.000   0.900  5.30 40.7  3.3  0.0  0.0  0.0  1.9  NA
## 36   84.460   3.600 18.30 12.4  1.0  0.0  0.0  0.0  1.0  NA
## 38    2.500   3.000  0.30 75.8  0.0  0.0  0.0  0.0  0.0 0.0
## 48    5.000   6.000  1.10 35.5  0.0  0.0  0.0  0.0  0.0 0.0
## 55    1.000   6.500    NA 24.3  0.0  0.0  0.0  0.0  0.0 0.0
## 56    1.000   1.000    NA 82.7  0.0  0.0  0.0  0.0  0.0 0.0
## 57    1.000   4.000    NA 16.8  4.6  3.9 11.5  0.0  0.0 0.0
## 58    1.000   6.000    NA 46.8  0.0  0.0 28.8  0.0  0.0 0.0
## 59    2.000  11.000    NA 46.9  0.0  0.0 13.4  0.0  0.0 0.0
## 60    1.000   6.000    NA 47.1  0.0  0.0  0.0  0.0  1.2 0.0
## 61    2.000  14.000    NA 66.9  0.0  0.0  0.0  0.0  0.0 0.0
## 62       NA  14.000    NA 19.4  0.0  0.0  2.0  0.0  3.9 1.7
## 63    3.333   6.667    NA 14.4  0.0  0.0  0.0  0.0  0.0 0.0
## 69  351.600  10.000  0.00  0.0  1.5  7.6  0.0  0.0  6.1  NA
```

```
## 70  313.600   1.000  1.90  4.9  2.6  3.0  0.0  0.0  1.9  NA
## 71  279.066  13.100 25.50  3.9  1.0 11.0  0.0  0.0 12.5  NA
## 88  607.167   4.300  0.00  0.0  2.6  2.4  5.0  0.0  2.4  NA
## 89  624.733   6.800  0.00  0.0  0.0  1.0 35.6  9.9  0.0  NA
## 116  22.444  10.111    NA 41.0  1.5  0.0  0.0  0.0  0.0 0.0
## 133 173.750  15.300  0.00  0.0  1.0  0.0  9.0 64.6  0.0  NA
## 146 276.000   8.100  6.50  4.1  0.0  7.7  9.9 18.2  7.0  NA
## 153  34.000  53.100  2.20  0.0  0.0  1.2  5.9 77.6  0.0  NA
## 156 361.000  28.567 24.80 10.4  0.0  6.9  0.0  0.0  2.7  NA
## 157 236.000  22.500 32.50 12.0  0.0  5.0  0.0  0.0  1.9  NA
## 161 102.000 186.000 68.05  1.7 20.6  1.5  2.2  0.0  0.0 0.0
## 171 272.222  20.578  2.50 13.2  0.0  2.0  7.4 17.2  0.0  NA
## 172 388.167   5.083  1.70 12.0  4.9  2.7  0.0  5.9  1.7  NA
## 184  21.083  56.091    NA 16.8 19.6  4.0  0.0  0.0  0.0 0.0
## 199      NA      NA    NA  0.0 12.5  3.7  1.0  0.0  0.0 4.9
```

```
nrow(algae[!complete.cases(algae),])
```

```
## [1] 33
```

```
#removes 16 water samples from the data frames
algae <- na.omit(algae)

algae <- algae[-c(62, 199), ]

#The following code gives you the number of unknown values in each row of the algae dataset
apply(algae, 1, function(x) sum(is.na(x)))
```

```
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##  19  22  23  24  25  26  27  29  30  31  32  33  37  39  40  41  42  43
##   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##  44  45  46  47  49  50  51  52  53  54  64  65  66  67  68  72  73  74
##   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##  75  76  77  78  79  80  81  83  84  85  86  87  90  91  92  93  94  95
##   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##  96  97  98  99 100 101 102 103 104 105 106 107 108 109 110 111 112 113
##   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 114 115 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132
##   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 134 135 136 137 138 139 140 141 142 143 144 145 147 148 149 150 151 152
##   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 154 155 158 159 160 162 163 164 165 166 167 168 169 170 173 174 175 176
##   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 177 178 179 180 181 182 183 185 186 187 188 189 190 191 192 193 194 195
##   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 196 197 198 200
##   0   0   0   0
```

```
#gives rows in the algae
data(algae)
#gives the row numbers having more than 20% of the columns with an NA
manyNAs(algae, 0.2)
```

```
## [1]  62 199
```

```
#alternative code
algae <- algae[-manyNAs(algae), ]

#since it is normal distribution, we use the mean value to fill in the hole
algae[48, "mxPH"] <- mean(algae$mxPH, na.rm = T)

#we use median, to fill in all the unknowns in this column
algae[is.na(algae$Chla), "Chla"] <- median(algae$Chla, na.rm = T)

#This function uses the median for numeric columns and uses the most frequent value (the mode) f
or nominal variables.
data(algae)
algae <- algae[-manyNAs(algae), ]
algae <- centralImputation(algae)

#to obtain the correlation of variables
#produces a matrix with the correlation values between the variables
cor(algae[, 4:18], use = "complete.obs")
```

```
##             mxPH        mnO2          Cl         NO3         NH4
## mxPH  1.00000000 -0.16749178  0.13285681 -0.13103951 -0.09360612
## mnO2 -0.16749178  1.00000000 -0.27873229  0.09837676 -0.08780541
## Cl    0.13285681 -0.27873229  1.00000000  0.22504071  0.07407466
## NO3  -0.13103951  0.09837676  0.22504071  1.00000000  0.72144352
## NH4  -0.09360612 -0.08780541  0.07407466  0.72144352  1.00000000
## oPO4  0.15850785 -0.41655069  0.39230733  0.14458782  0.22723723
## PO4   0.18033494 -0.48772564  0.45652107  0.16931401  0.20844445
## Chla  0.39121495 -0.16678069  0.15082753  0.14290962  0.09375115
## a1   -0.26823725  0.28389830 -0.36078101 -0.24121109 -0.13265601
## a2    0.32584814 -0.09935631  0.08949837  0.02368832 -0.02968344
## a3    0.03077250 -0.25155437  0.09429722 -0.07621407 -0.10143974
## a4   -0.24876290 -0.31513753  0.12045912 -0.02578257  0.22822914
## a5   -0.01697947  0.17008979  0.16514900  0.22359794  0.02745909
## a6   -0.08388657  0.15864906  0.18369968  0.54640569  0.40571045
## a7   -0.08726106 -0.12117098 -0.02793640  0.08509789 -0.01672691
##             oPO4         PO4        Chla          a1          a2
## mxPH  0.15850785  0.18033494  0.39121495 -0.26823725  0.32584814
## mnO2 -0.41655069 -0.48772564 -0.16678069  0.28389830 -0.09935631
## Cl    0.39230733  0.45652107  0.15082753 -0.36078101  0.08949837
## NO3   0.14458782  0.16931401  0.14290962 -0.24121109  0.02368832
## NH4   0.22723723  0.20844445  0.09375115 -0.13265601 -0.02968344
## oPO4  1.00000000  0.91387767  0.12941615 -0.41735761  0.14768993
## PO4   0.91387767  1.00000000  0.26758873 -0.48730097  0.16246963
## Chla  0.12941615  0.26758873  1.00000000 -0.28380049  0.38192280
## a1   -0.41735761 -0.48730097 -0.28380049  1.00000000 -0.29251967
## a2    0.14768993  0.16246963  0.38192280 -0.29251967  1.00000000
## a3    0.03362906  0.06587312 -0.04975884 -0.14695028  0.03031095
## a4    0.29574585  0.30462623 -0.08364618 -0.03892441 -0.17168171
## a5    0.15147500  0.19111521 -0.05945318 -0.29503346 -0.16186215
## a6    0.02876159  0.08316987  0.01815732 -0.27602608 -0.11613061
## a7    0.04849832  0.10671057  0.02405581 -0.21142489  0.04749242
##              a3          a4          a5          a6          a7
## mxPH  0.03077250 -0.24876290 -0.01697947 -0.08388657 -0.08726106
## mnO2 -0.25155437 -0.31513753  0.17008979  0.15864906 -0.12117098
## Cl    0.09429722  0.12045912  0.16514900  0.18369968 -0.02793640
## NO3  -0.07621407 -0.02578257  0.22359794  0.54640569  0.08509789
## NH4  -0.10143974  0.22822914  0.02745909  0.40571045 -0.01672691
## oPO4  0.03362906  0.29574585  0.15147500  0.02876159  0.04849832
## PO4   0.06587312  0.30462623  0.19111521  0.08316987  0.10671057
## Chla -0.04975884 -0.08364618 -0.05945318  0.01815732  0.02405581
## a1   -0.14695028 -0.03892441 -0.29503346 -0.27602608 -0.21142489
## a2    0.03031095 -0.17168171 -0.16186215 -0.11613061  0.04749242
## a3    1.00000000  0.01218370 -0.11111997 -0.17283566  0.05618729
## a4    0.01218370  1.00000000 -0.11006558 -0.09074936  0.04362334
## a5   -0.11111997 -0.11006558  1.00000000  0.40360881 -0.02686306
## a6   -0.17283566 -0.09074936  0.40360881  1.00000000 -0.01244488
## a7    0.05618729  0.04362334 -0.02686306 -0.01244488  1.00000000
```

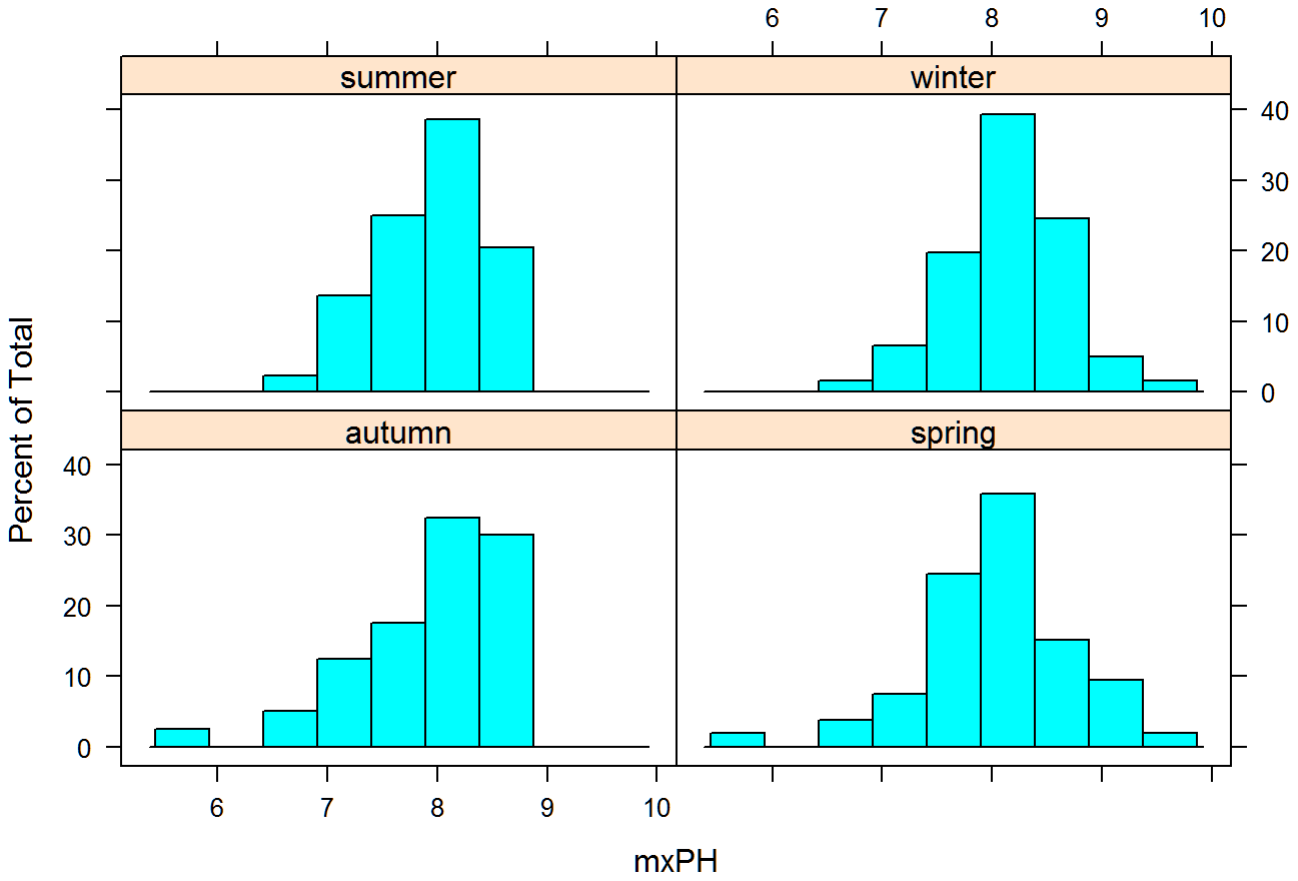
```
#to make it more legible, we use "symnum" function
symnum(cor(algae[,4:18],use="complete.obs"))
```

```
##      mP mO Cl NO NH o P Ch a1 a2 a3 a4 a5 a6 a7
## mxPH 1
## mnO2    1
## Cl         1
## NO3           1
## NH4           ,  1
## oPO4    .  .        1
## PO4     .  .        * 1
## Chla .                  1
## a1          .        . .    1
## a2   .                  .     1
## a3                             1
## a4      .             .            1
## a5                                    1
## a6            .  .                    .  1
## a7                                          1
## attr(,"legend")
## [1] 0 ' ' 0.3 '.' 0.6 ',' 0.8 '+' 0.9 '*' 0.95 'B' 1
```

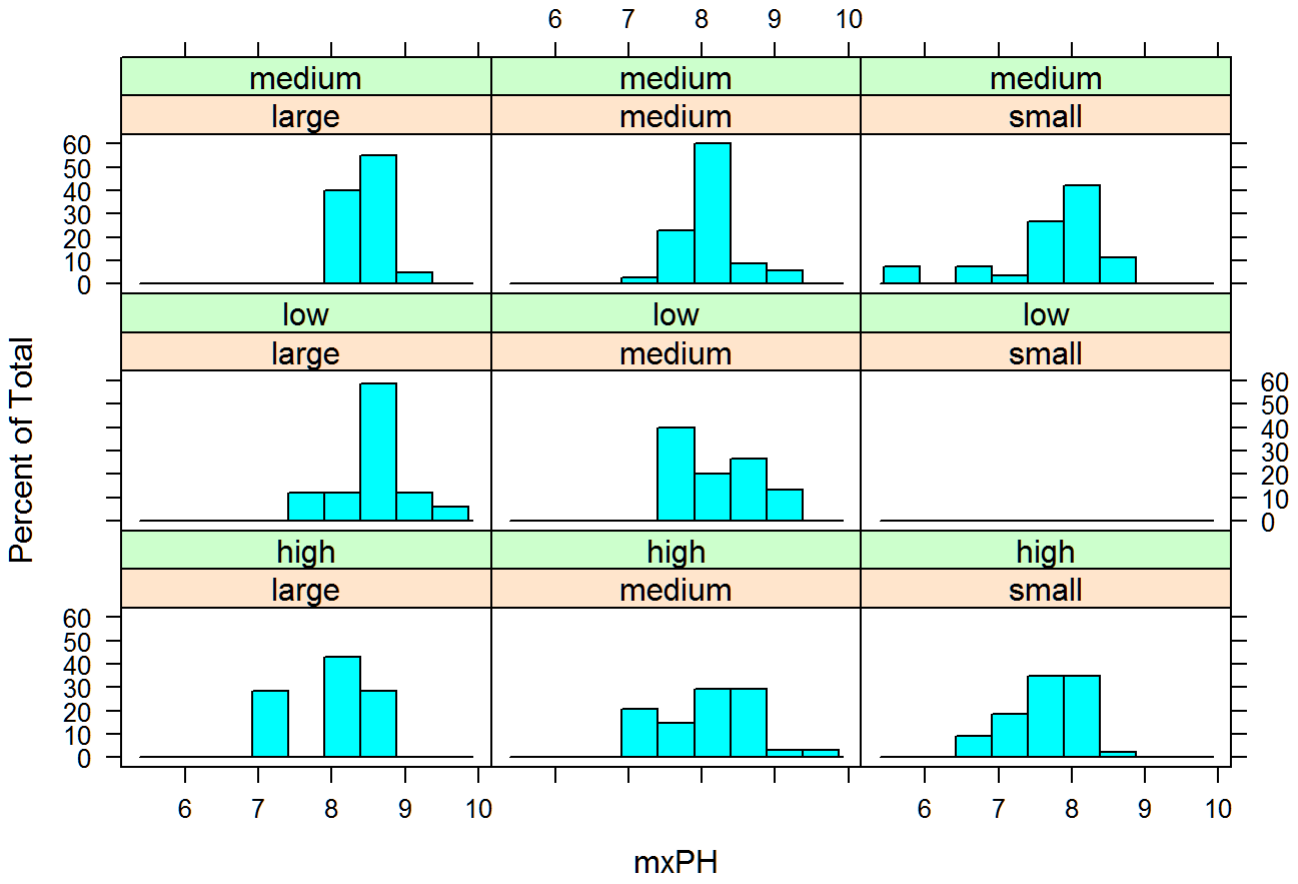
```
data(algae)
algae <- algae[-manyNAs(algae), ]
#to obtain linear models
# we can fill in the unknown values of these variables
lm(PO4 ~ oPO4, data = algae)
```

```
##
## Call:
## lm(formula = PO4 ~ oPO4, data = algae)
##
## Coefficients:
## (Intercept)         oPO4
##      42.897        1.293
```

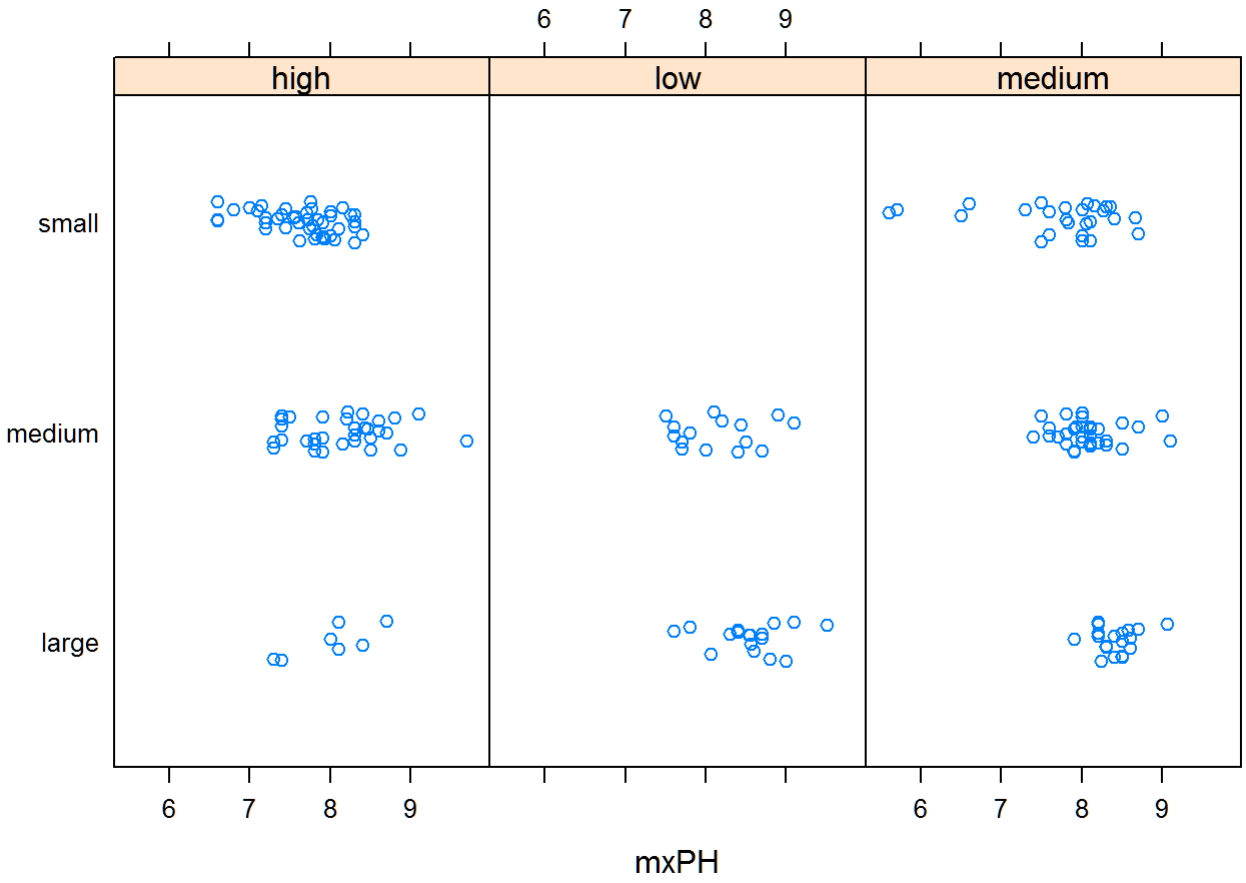
```
algae[28, "PO4"] <- 42.897 + 1.293 * algae[28, "oPO4"]

#uses above linear relationship to fill in all the unknowns
data(algae)
algae <- algae[-manyNAs(algae), ]

# this function returns the value of PO4 according to the discovered linear relation
fillPO4 <- function(oP) {
  if (is.na(oP))
    return(NA)
  else return(42.897 + 1.293 * oP)
}


algae[is.na(algae$PO4), "PO4"] <- sapply(algae[is.na(algae$PO4),
                                               "oPO4"], fillPO4)

##################################################################

# obtains an histogram of the values of mxPH for the di???erent values of season. Each histogram
 is built using only the subset of observations with a certain season value
histogram(~mxPH | season, data = algae)
```



```
#changes the ordering of the labels that form the factor season in the data frame.
algae$season <- factor(algae$season, levels = c("spring",
                                                "summer", "autumn", "winter"))

#plots a histogram and shows the variation of mxPH for all combinations of size and speed of the
 rivers.
histogram(~mxPH | size * speed, data = algae)
```




```
#Another alternative to obtain similar information but now with the concrete values of the varia
ble
stripplot(size ~ mxPH | speed, data = algae, jitter = T)
```




```
##############################################################

#
data(algae)
algae <- algae[-manyNAs(algae), ]

algae <- knnImputation(algae, k = 10)

# uses the median values for filling in the unknowns
algae <- knnImputation(algae, k = 10, meth = "median")
```

```
## Warning in knnImputation(algae, k = 10, meth = "median"): No case has
## missing values. Stopping as there is nothing to do.
```

```
############################################################
data(algae)
algae <- algae[-manyNAs(algae), ]
clean.algae <- knnImputation(algae, k = 10)

lm.a1 <- lm(a1 ~ ., data = clean.algae[, 1:12])

#gives some diagnostic information concerning the obtained model.
#creates three auxilliary variables for the factor season
#This means that if we have a water sample with the value "autumn" in the variable season, all t
hree auxiliary variables will be set to zero.
summary(lm.a1)
```

```
##
## Call:
## lm(formula = a1 ~ ., data = clean.algae[, 1:12])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -37.679 -11.893  -2.567   7.410  62.190
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  42.942055  24.010879   1.788  0.07537 .
## seasonspring  3.726978   4.137741   0.901  0.36892
## seasonsummer  0.747597   4.020711   0.186  0.85270
## seasonwinter  3.692955   3.865391   0.955  0.34065
## sizemedium    3.263728   3.802051   0.858  0.39179
## sizesmall     9.682140   4.179971   2.316  0.02166 *
## speedlow      3.922084   4.706315   0.833  0.40573
## speedmedium   0.246764   3.241874   0.076  0.93941
## mxPH         -3.589118   2.703528  -1.328  0.18598
## mnO2          1.052636   0.705018   1.493  0.13715
## Cl           -0.040172   0.033661  -1.193  0.23426
## NO3          -1.511235   0.551339  -2.741  0.00674 **
## NH4           0.001634   0.001003   1.628  0.10516
## oPO4         -0.005435   0.039884  -0.136  0.89177
## PO4          -0.052241   0.030755  -1.699  0.09109 .
## Chla         -0.088022   0.079998  -1.100  0.27265
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.65 on 182 degrees of freedom
## Multiple R-squared:  0.3731, Adjusted R-squared:  0.3215
## F-statistic: 7.223 on 15 and 182 DF,  p-value: 2.444e-12
```

```
#indicates that the variable season is the variable that least contributes to the reduction of t
he fitting error of the model.
anova(lm.a1)
```

```
## Analysis of Variance Table
##
## Response: a1
##            Df Sum Sq Mean Sq F value    Pr(>F)
## season      3     85    28.2  0.0905 0.9651944
## size        2  11401  5700.7 18.3088  5.69e-08 ***
## speed       2   3934  1967.2  6.3179 0.0022244 **
## mxPH        1   1329  1328.8  4.2677 0.0402613 *
## mnO2        1   2287  2286.8  7.3444 0.0073705 **
## Cl          1   4304  4304.3 13.8239 0.0002671 ***
## NO3         1   3418  3418.5 10.9789 0.0011118 **
## NH4         1    404   403.6  1.2963 0.2563847
## oPO4        1   4788  4788.0 15.3774 0.0001246 ***
## PO4         1   1406  1405.6  4.5142 0.0349635 *
## Chla        1    377   377.0  1.2107 0.2726544
## Residuals 182  56668   311.4
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#to obtain a new model by removing the variable season from the lm.a1 model.
lm2.a1 <- update(lm.a1, . ~ . - season)

#prints the summmary information for the new model
#The fit has improved a bit (32.8%) but it is still not too impressive
summary(lm2.a1)
```

```
##
## Call:
## lm(formula = a1 ~ size + speed + mxPH + mnO2 + Cl + NO3 + NH4 +
##     oPO4 + PO4 + Chla, data = clean.algae[, 1:12])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -36.460 -11.953  -3.044   7.444  63.730
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 44.9532874 23.2378377   1.934  0.05458 .
## sizemedium   3.3092102  3.7825221   0.875  0.38278
## sizesmall   10.2730961  4.1223163   2.492  0.01358 *
## speedlow     3.0546270  4.6108069   0.662  0.50848
## speedmedium -0.2976867  3.1818585  -0.094  0.92556
## mxPH        -3.2684281  2.6576592  -1.230  0.22033
## mnO2         0.8011759  0.6589644   1.216  0.22561
## Cl          -0.0381881  0.0333791  -1.144  0.25407
## NO3         -1.5334300  0.5476550  -2.800  0.00565 **
## NH4          0.0015777  0.0009951   1.586  0.11456
## oPO4        -0.0062392  0.0395086  -0.158  0.87469
## PO4         -0.0509543  0.0305189  -1.670  0.09669 .
## Chla        -0.0841371  0.0794459  -1.059  0.29096
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.57 on 185 degrees of freedom
## Multiple R-squared:  0.3682, Adjusted R-squared:  0.3272
## F-statistic: 8.984 on 12 and 185 DF,  p-value: 1.762e-13
```

```
#comparison between the two models by using again the anova() function.
anova(lm.a1,lm2.a1)
```

```
## Analysis of Variance Table
##
## Model 1: a1 ~ season + size + speed + mxPH + mnO2 + Cl + NO3 + NH4 + oPO4 +
##     PO4 + Chla
## Model 2: a1 ~ size + speed + mxPH + mnO2 + Cl + NO3 + NH4 + oPO4 + PO4 +
##     Chla
##   Res.Df   RSS Df Sum of Sq      F Pr(>F)
## 1    182 56668
## 2    185 57116 -3   -447.62 0.4792 0.6971
```

```
#The following code creates a linear model that results from applying the backward elimination m
ethod to the initial model
final.lm <- step(lm.a1)
```

```
## Start:  AIC=1152.03
## a1 ~ season + size + speed + mxPH + mnO2 + Cl + NO3 + NH4 + oPO4 +
##     PO4 + Chla
##
##          Df Sum of Sq   RSS    AIC
## - season  3    447.62 57116 1147.6
## - speed   2    269.60 56938 1149.0
## - oPO4    1      5.78 56674 1150.0
## - Chla    1    376.96 57045 1151.3
## - Cl      1    443.46 57112 1151.6
## - mxPH    1    548.76 57217 1151.9
## <none>                56668 1152.0
## - mnO2    1    694.11 57363 1152.4
## - NH4     1    825.67 57494 1152.9
## - PO4     1    898.42 57567 1153.1
## - size    2   1857.16 58526 1154.4
## - NO3     1   2339.36 59008 1158.0
##
## Step:  AIC=1147.59
## a1 ~ size + speed + mxPH + mnO2 + Cl + NO3 + NH4 + oPO4 + PO4 +
##     Chla
##
##         Df Sum of Sq   RSS    AIC
## - speed  2    210.64 57327 1144.3
## - oPO4   1      7.70 57124 1145.6
## - Chla   1    346.27 57462 1146.8
## - Cl     1    404.10 57520 1147.0
## - mnO2   1    456.37 57572 1147.2
## - mxPH   1    466.95 57583 1147.2
## <none>               57116 1147.6
## - NH4    1    776.11 57892 1148.3
## - PO4    1    860.62 57977 1148.5
## - size   2   2175.59 59292 1151.0
## - NO3    1   2420.47 59537 1153.8
##
## Step:  AIC=1144.31
## a1 ~ size + mxPH + mnO2 + Cl + NO3 + NH4 + oPO4 + PO4 + Chla
##
##        Df Sum of Sq   RSS    AIC
## - oPO4  1     16.29 57343 1142.4
## - Chla  1    223.29 57550 1143.1
## - mnO2  1    413.77 57740 1143.7
## - Cl    1    472.70 57799 1143.9
## - mxPH  1    483.56 57810 1144.0
## <none>              57327 1144.3
## - NH4   1    720.19 58047 1144.8
## - PO4   1    809.30 58136 1145.1
## - size  2   2060.95 59388 1147.3
## - NO3   1   2379.75 59706 1150.4
##
## Step:  AIC=1142.37
## a1 ~ size + mxPH + mnO2 + Cl + NO3 + NH4 + PO4 + Chla
##
```

```
##        Df Sum of Sq   RSS    AIC
## - Chla  1     207.7 57551 1141.1
## - mnO2  1     402.6 57746 1141.8
## - Cl    1     470.7 57814 1142.0
## - mxPH  1     519.7 57863 1142.2
## <none>              57343 1142.4
## - NH4   1     704.4 58047 1142.8
## - size  2    2050.3 59393 1145.3
## - NO3   1    2370.4 59713 1148.4
## - PO4   1    5818.4 63161 1159.5
##
## Step:  AIC=1141.09
## a1 ~ size + mxPH + mnO2 + Cl + NO3 + NH4 + PO4
##
##        Df Sum of Sq   RSS    AIC
## - mnO2  1     435.3 57986 1140.6
## - Cl    1     438.1 57989 1140.6
## <none>              57551 1141.1
## - NH4   1     746.9 58298 1141.6
## - mxPH  1     833.1 58384 1141.9
## - size  2    2217.5 59768 1144.6
## - NO3   1    2667.1 60218 1148.1
## - PO4   1    6309.7 63860 1159.7
##
## Step:  AIC=1140.58
## a1 ~ size + mxPH + Cl + NO3 + NH4 + PO4
##
##        Df Sum of Sq   RSS    AIC
## - NH4   1     531.0 58517 1140.4
## - Cl    1     584.9 58571 1140.6
## <none>              57986 1140.6
## - mxPH  1     819.1 58805 1141.4
## - size  2    2478.2 60464 1144.9
## - NO3   1    2251.4 60237 1146.1
## - PO4   1    9097.9 67084 1167.4
##
## Step:  AIC=1140.38
## a1 ~ size + mxPH + Cl + NO3 + PO4
##
##        Df Sum of Sq   RSS    AIC
## <none>              58517 1140.4
## - mxPH  1     784.1 59301 1141.0
## - Cl    1     835.6 59353 1141.2
## - NO3   1    1987.9 60505 1145.0
## - size  2    2664.3 61181 1145.2
## - PO4   1    8575.8 67093 1165.5
```

```
# obtain the information on the final model
summary(final.lm)
```

```
##
## Call:
## lm(formula = a1 ~ size + mxPH + Cl + NO3 + PO4, data = clean.algae[,
##     1:12])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.874 -12.732  -3.741   8.424  62.926
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 57.28555   20.96132   2.733  0.00687 **
## sizemedium   2.80050    3.40190   0.823  0.41141
## sizesmall   10.40636    3.82243   2.722  0.00708 **
## mxPH        -3.97076    2.48204  -1.600  0.11130
## Cl          -0.05227    0.03165  -1.651  0.10028
## NO3         -0.89529    0.35148  -2.547  0.01165 *
## PO4         -0.05911    0.01117  -5.291 3.32e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.5 on 191 degrees of freedom
## Multiple R-squared:  0.3527, Adjusted R-squared:  0.3324
## F-statistic: 17.35 on 6 and 191 DF,  p-value: 5.554e-16
```

```
#Regression trees

#Loads the library "rpart"
library(rpart)
```

```
##
## Attaching package: 'rpart'
```

```
## The following object is masked from 'package:survival':
##
##     solder
```

```
data(algae)
algae <- algae[-manyNAs(algae), ]
rt.a1 <- rpart(a1 ~ ., data = algae[, 1:12])

#displays the content of the object "rt.a1"
rt.a1
```

```
## n= 198
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 198 90401.290 16.996460
##    2) PO4>=43.818 147 31279.120  8.979592
##      4) Cl>=7.8065 140 21622.830  7.492857
##        8) oPO4>=51.118 84  3441.149  3.846429 *
##        9) oPO4< 51.118 56 15389.430 12.962500
##         18) mnO2>=10.05 24  1248.673  6.716667 *
##         19) mnO2< 10.05 32 12502.320 17.646870
##           38) NO3>=3.1875 9   257.080  7.866667 *
##           39) NO3< 3.1875 23 11047.500 21.473910
##             78) mnO2< 8 13  2919.549 13.807690 *
##             79) mnO2>=8 10  6370.704 31.440000 *
##      5) Cl< 7.8065 7  3157.769 38.714290 *
##    3) PO4< 43.818 51 22442.760 40.103920
##      6) mxPH< 7.87 28 11452.770 33.450000
##       12) mxPH>=7.045 18  5146.169 26.394440 *
##       13) mxPH< 7.045 10  3797.645 46.150000 *
##      7) mxPH>=7.87 23  8241.110 48.204350
##       14) PO4>=15.177 12  3047.517 38.183330 *
##       15) PO4< 15.177 11  2673.945 59.136360 *
```

```
#obtain a graphical representation of the tree
prettyTree(rt.a1)
#This will produce a lot of information concerning the tests on the tree, the alternative tests
 that could be considered, and also the surrogate splits.
summary(rt.a1)
```

```
## Call:
## rpart(formula = a1 ~ ., data = algae[, 1:12])
##   n= 198
##
##           CP nsplit rel error    xerror      xstd
## 1 0.40573990      0 1.0000000 1.0033849 0.1298582
## 2 0.07188523      1 0.5942601 0.7688237 0.1209498
## 3 0.03088731      2 0.5223749 0.7224747 0.1195087
## 4 0.03040753      3 0.4914876 0.7357163 0.1205986
## 5 0.02787181      4 0.4610800 0.7346177 0.1215271
## 6 0.02775354      5 0.4332082 0.7570904 0.1252944
## 7 0.01812406      6 0.4054547 0.7651926 0.1226123
## 8 0.01634372      7 0.3873306 0.7110891 0.1158854
## 9 0.01000000      9 0.3546432 0.7171105 0.1191155
##
## Variable importance
##    PO4   oPO4    NH4     Cl   mxPH   Chla    NO3   mnO2   size season
##     25     20     15     15      9      7      3      2      1      1
##  speed
##      1
##
## Node number 1: 198 observations,    complexity param=0.4057399
##   mean=16.99646, MSE=456.5722
##   left son=2 (147 obs) right son=3 (51 obs)
##   Primary splits:
##       PO4  < 43.818   to the right, improve=0.4048567, (1 missing)
##       oPO4 < 18.889   to the right, improve=0.3793450, (0 missing)
##       NH4  < 51.27    to the right, improve=0.3625269, (0 missing)
##       Cl   < 7.2915   to the right, improve=0.3583409, (8 missing)
##       Chla < 1.15     to the right, improve=0.2533869, (10 missing)
##   Surrogate splits:
##       oPO4 < 17.5415  to the right, agree=0.944, adj=0.78, (1 split)
##       NH4  < 37.639   to the right, agree=0.893, adj=0.58, (0 split)
##       Cl   < 9.0275   to the right, agree=0.858, adj=0.44, (0 split)
##       Chla < 1.05     to the right, agree=0.822, adj=0.30, (0 split)
##       mxPH < 7.295    to the right, agree=0.817, adj=0.28, (0 split)
##
## Node number 2: 147 observations,    complexity param=0.07188523
##   mean=8.979592, MSE=212.7831
##   left son=4 (140 obs) right son=5 (7 obs)
##   Primary splits:
##       Cl   < 7.8065   to the right, improve=0.2071337, (1 missing)
##       Chla < 1.15     to the right, improve=0.1959676, (1 missing)
##       oPO4 < 51.118   to the right, improve=0.1651094, (0 missing)
##       NH4  < 49.25    to the right, improve=0.1494842, (0 missing)
##       PO4  < 125      to the right, improve=0.1393822, (0 missing)
##   Surrogate splits:
##       Chla < 0.6      to the right, agree=0.959, adj=0.143, (1 split)
##
## Node number 3: 51 observations,    complexity param=0.03040753
##   mean=40.10392, MSE=440.0541
##   left son=6 (28 obs) right son=7 (23 obs)
##   Primary splits:
```

```
##       mxPH < 7.87     to the left,  improve=0.12171490, (1 missing)
##       PO4  < 6.35     to the right, improve=0.10576260, (1 missing)
##       Cl   < 7.544    to the right, improve=0.10428070, (7 missing)
##       NH4  < 18.381   to the right, improve=0.10356000, (0 missing)
##       oPO4 < 10.625   to the right, improve=0.09644168, (0 missing)
##   Surrogate splits:
##       size   splits as  RRL,          agree=0.78, adj=0.522, (1 split)
##       NO3    < 1.1875   to the right, agree=0.74, adj=0.435, (0 split)
##       oPO4   < 3.111    to the left,  agree=0.70, adj=0.348, (0 split)
##       season splits as  LLRR,         agree=0.60, adj=0.130, (0 split)
##       NH4    < 22.0355  to the left,  agree=0.60, adj=0.130, (0 split)
##
## Node number 4: 140 observations,    complexity param=0.03088731
##   mean=7.492857, MSE=154.4488
##   left son=8 (84 obs) right son=9 (56 obs)
##   Primary splits:
##       oPO4 < 51.118   to the right, improve=0.12913450, (0 missing)
##       PO4  < 125      to the right, improve=0.09908251, (0 missing)
##       NH4  < 41.875   to the right, improve=0.05847356, (0 missing)
##       NO3  < 3.2725   to the right, improve=0.05343570, (0 missing)
##       Chla < 3.65     to the right, improve=0.04761161, (1 missing)
##   Surrogate splits:
##       PO4    < 125      to the right, agree=0.857, adj=0.643, (0 split)
##       Cl     < 27.8665  to the right, agree=0.721, adj=0.304, (0 split)
##       NO3    < 3.313    to the right, agree=0.679, adj=0.196, (0 split)
##       mnO2   < 9.5      to the left,  agree=0.664, adj=0.161, (0 split)
##       season splits as  RLLL,         agree=0.657, adj=0.143, (0 split)
##
## Node number 5: 7 observations
##   mean=38.71429, MSE=451.1098
##
## Node number 6: 28 observations,    complexity param=0.02775354
##   mean=33.45, MSE=409.0275
##   left son=12 (18 obs) right son=13 (10 obs)
##   Primary splits:
##       mxPH   < 7.045    to the right, improve=0.2296931, (1 missing)
##       PO4    < 6.25     to the right, improve=0.2174386, (1 missing)
##       oPO4   < 12.375   to the right, improve=0.1721865, (0 missing)
##       NH4    < 17.1     to the right, improve=0.1098949, (0 missing)
##       season splits as  LRRR,         improve=0.0944271, (0 missing)
##   Surrogate splits:
##       NH4   < 11.25    to the right, agree=0.852, adj=0.556, (1 split)
##       oPO4  < 1.125    to the right, agree=0.852, adj=0.556, (0 split)
##       PO4   < 6.5835   to the right, agree=0.852, adj=0.556, (0 split)
##       speed splits as  L-R,          agree=0.778, adj=0.333, (0 split)
##       NO3   < 1.9675   to the left,  agree=0.778, adj=0.333, (0 split)
##
## Node number 7: 23 observations,    complexity param=0.02787181
##   mean=48.20435, MSE=358.3091
##   left son=14 (12 obs) right son=15 (11 obs)
##   Primary splits:
##       PO4  < 15.177   to the right, improve=0.3057413, (0 missing)
##       NH4  < 20.4165  to the right, improve=0.2692864, (0 missing)
##       Cl   < 7.544    to the right, improve=0.2055829, (0 missing)
```

```
##       Chla < 0.85     to the right, improve=0.1534699, (1 missing)
##       oPO4 < 6.25     to the right, improve=0.1013330, (0 missing)
##   Surrogate splits:
##       NH4  < 20.4165  to the right, agree=0.913, adj=0.818, (0 split)
##       Cl   < 5.8595   to the right, agree=0.826, adj=0.636, (0 split)
##       NO3  < 1.353    to the right, agree=0.826, adj=0.636, (0 split)
##       oPO4 < 5        to the right, agree=0.783, adj=0.545, (0 split)
##       Chla < 0.85     to the right, agree=0.739, adj=0.455, (0 split)
##
## Node number 8: 84 observations
##   mean=3.846429, MSE=40.96606
##
## Node number 9: 56 observations,    complexity param=0.01812406
##   mean=12.9625, MSE=274.8113
##   left son=18 (24 obs) right son=19 (32 obs)
##   Primary splits:
##       mnO2 < 10.05    to the right, improve=0.10646520, (0 missing)
##       PO4  < 101.894  to the left,  improve=0.08815216, (0 missing)
##       oPO4 < 24.3335  to the left,  improve=0.07637520, (0 missing)
##       size splits as  LLR,          improve=0.06017653, (0 missing)
##       mxPH < 8.35     to the right, improve=0.05440345, (0 missing)
##   Surrogate splits:
##       PO4    < 101.894  to the left,  agree=0.750, adj=0.417, (0 split)
##       size   splits as  LRR,          agree=0.696, adj=0.292, (0 split)
##       season splits as  LRRR,         agree=0.679, adj=0.250, (0 split)
##       NH4    < 89.8     to the left,  agree=0.661, adj=0.208, (0 split)
##       mxPH   < 8.025    to the right, agree=0.643, adj=0.167, (0 split)
##
## Node number 12: 18 observations
##   mean=26.39444, MSE=285.8983
##
## Node number 13: 10 observations
##   mean=46.15, MSE=379.7645
##
## Node number 14: 12 observations
##   mean=38.18333, MSE=253.9597
##
## Node number 15: 11 observations
##   mean=59.13636, MSE=243.086
##
## Node number 18: 24 observations
##   mean=6.716667, MSE=52.02806
##
## Node number 19: 32 observations,    complexity param=0.01634372
##   mean=17.64687, MSE=390.6975
##   left son=38 (9 obs) right son=39 (23 obs)
##   Primary splits:
##       NO3  < 3.1875   to the right, improve=0.09580105, (0 missing)
##       Chla < 2.55     to the left,  improve=0.08399898, (0 missing)
##       oPO4 < 24.917   to the left,  improve=0.07524892, (0 missing)
##       mnO2 < 9.4      to the left,  improve=0.06578127, (0 missing)
##       Cl   < 43.7085  to the right, improve=0.04807023, (0 missing)
##   Surrogate splits:
##       mxPH < 7.55     to the left,  agree=0.844, adj=0.444, (0 split)
```

```
##       NH4  < 224.643  to the right, agree=0.812, adj=0.333, (0 split)
##       PO4  < 206.7225 to the right, agree=0.812, adj=0.333, (0 split)
##       Cl   < 84.0465  to the right, agree=0.750, adj=0.111, (0 split)
##
## Node number 38: 9 observations
##   mean=7.866667, MSE=28.56444
##
## Node number 39: 23 observations,    complexity param=0.01634372
##   mean=21.47391, MSE=480.3263
##   left son=78 (13 obs) right son=79 (10 obs)
##   Primary splits:
##       mnO2 < 8        to the left,  improve=0.15906320, (0 missing)
##       PO4  < 118.6    to the left,  improve=0.10091960, (0 missing)
##       NH4  < 168.75   to the left,  improve=0.07651249, (0 missing)
##       NO3  < 1.2495   to the left,  improve=0.07260629, (0 missing)
##       mxPH < 8.26     to the right, improve=0.06930695, (0 missing)
##   Surrogate splits:
##       season splits as  RLLL,         agree=0.696, adj=0.3, (0 split)
##       size   splits as  LLR,          agree=0.696, adj=0.3, (0 split)
##       speed  splits as  RLL,          agree=0.696, adj=0.3, (0 split)
##       Cl     < 46.9725  to the right, agree=0.696, adj=0.3, (0 split)
##       NH4    < 216.653  to the left,  agree=0.696, adj=0.3, (0 split)
##
## Node number 78: 13 observations
##   mean=13.80769, MSE=224.5807
##
## Node number 79: 10 observations
##   mean=31.44, MSE=637.0704
```

```
#produces a set of sub-trees of this tree and estimate their predictive performance
printcp(rt.a1)
```

```
##
## Regression tree:
## rpart(formula = a1 ~ ., data = algae[, 1:12])
##
## Variables actually used in tree construction:
## [1] Cl   mnO2 mxPH NO3  oPO4 PO4
##
## Root node error: 90401/198 = 456.57
##
## n= 198
##
##         CP nsplit rel error  xerror    xstd
## 1 0.405740      0   1.00000 1.00338 0.12986
## 2 0.071885      1   0.59426 0.76882 0.12095
## 3 0.030887      2   0.52237 0.72247 0.11951
## 4 0.030408      3   0.49149 0.73572 0.12060
## 5 0.027872      4   0.46108 0.73462 0.12153
## 6 0.027754      5   0.43321 0.75709 0.12529
## 7 0.018124      6   0.40545 0.76519 0.12261
## 8 0.016344      7   0.38733 0.71109 0.11589
## 9 0.010000      9   0.35464 0.71711 0.11912
```

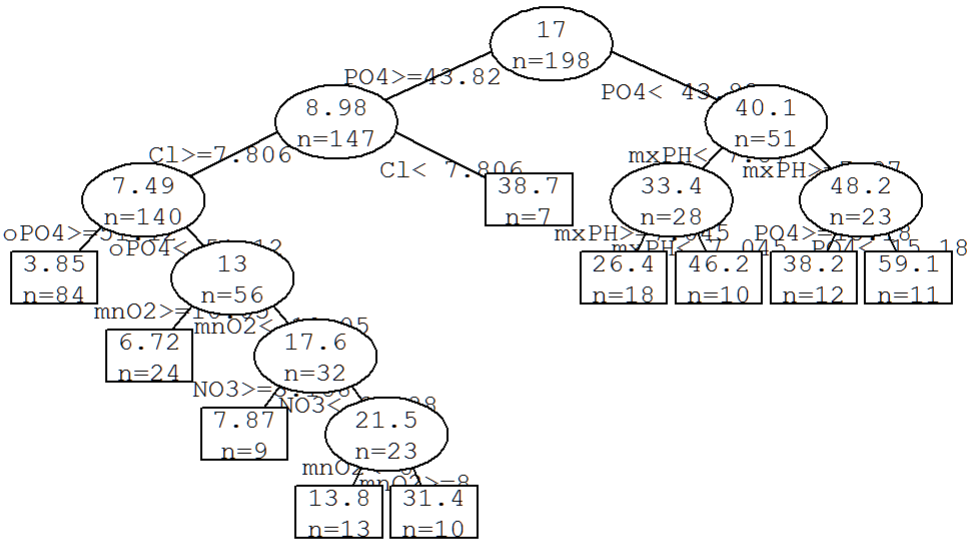
```
rt2.a1 <- prune(rt.a1, cp = 0.08)
rt2.a1
```

```
## n= 198
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
## 1) root 198 90401.29 16.996460
##   2) PO4>=43.818 147 31279.12  8.979592 *
##   3) PO4< 43.818 51 22442.76 40.103920 *
```

```
(rt.a1 <- rpartXse(a1 ~ ., data = algae[, 1:12]))
```

```
## n= 198
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
## 1) root 198 90401.290 16.996460
##   2) PO4>=43.818 147 31279.120  8.979592
##     4) Cl>=7.1665 142 21763.160  7.530282 *
##     5) Cl< 7.1665 5   746.792 50.140000 *
##   3) PO4< 43.818 51 22442.760 40.103920 *
```

```
#indicates the number of the nodes at which you want to prune the tree
first.tree <- rpart(a1 ~ ., data = algae[, 1:12])
my.tree <- snip.rpart(first.tree, c(4, 7))

prettyTree(first.tree)
snip.rpart(first.tree)
```



```
## n= 198
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 198 90401.290 16.996460
##    2) PO4>=43.818 147 31279.120  8.979592
##      4) Cl>=7.8065 140 21622.830  7.492857
##        8) oPO4>=51.118 84  3441.149  3.846429 *
##        9) oPO4< 51.118 56 15389.430 12.962500
##         18) mnO2>=10.05 24  1248.673  6.716667 *
##         19) mnO2< 10.05 32 12502.320 17.646870
##           38) NO3>=3.1875 9   257.080  7.866667 *
##           39) NO3< 3.1875 23 11047.500 21.473910
##             78) mnO2< 8 13  2919.549 13.807690 *
##             79) mnO2>=8 10  6370.704 31.440000 *
##      5) Cl< 7.8065 7  3157.769 38.714290 *
##    3) PO4< 43.818 51 22442.760 40.103920
##      6) mxPH< 7.87 28 11452.770 33.450000
##       12) mxPH>=7.045 18  5146.169 26.394440 *
##       13) mxPH< 7.045 10  3797.645 46.150000 *
##      7) mxPH>=7.87 23  8241.110 48.204350
##       14) PO4>=15.177 12  3047.517 38.183330 *
##       15) PO4< 15.177 11  2673.945 59.136360 *
```

```
#receives a model and a test dataset and retrieves the correspondent model predictions:
lm.predictions.a1 <- predict(final.lm, clean.algae)
rt.predictions.a1 <- predict(rt.a1, algae)

# calculates their mean absolute error
(mae.a1.lm <- mean(abs(lm.predictions.a1 - algae[, "a1"])))
```

```
## [1] 13.10681
```

```
(mae.a1.rt <- mean(abs(rt.predictions.a1 - algae[, "a1"])))
```

```
## [1] 10.36242
```

```
(mse.a1.lm <- mean((lm.predictions.a1 - algae[, "a1"])^2))
```

```
## [1] 295.5407
```

```
(mse.a1.rt <- mean((rt.predictions.a1 - algae[, "a1"])^2))
```

```
## [1] 227.0339
```

```
(nmse.a1.lm <- mean((lm.predictions.a1-algae[,'a1'])^2)/
    mean((mean(algae[,'a1'])-algae[,'a1'])^2))
```

```
## [1] 0.6473034
```

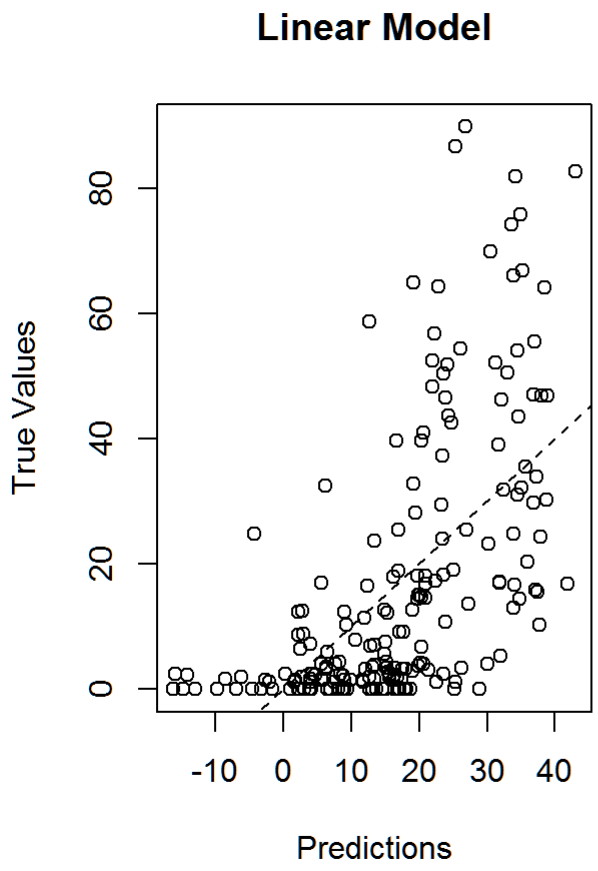
```
(nmse.a1.rt <- mean((rt.predictions.a1-algae[,'a1'])^2)/
    mean((mean(algae[,'a1'])-algae[,'a1'])^2))
```

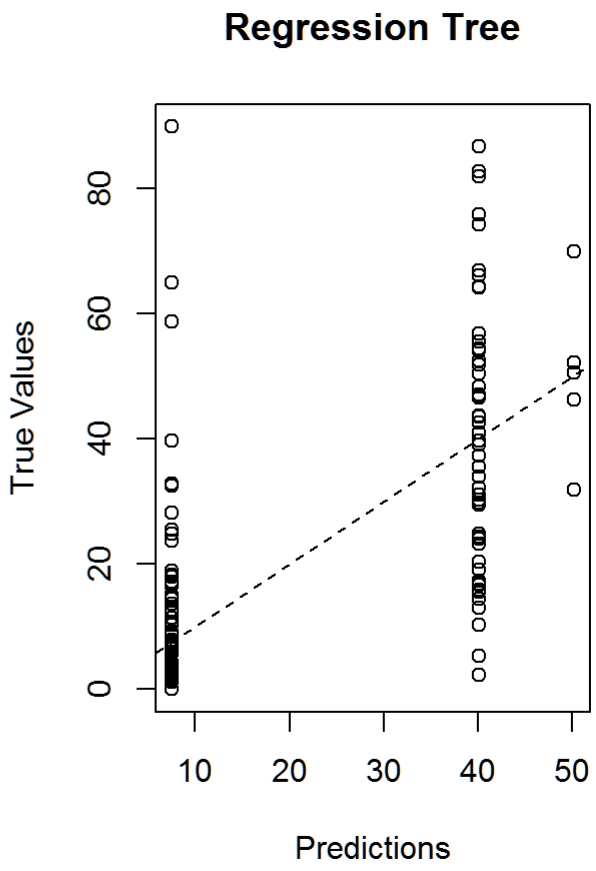
```
## [1] 0.4972574
```

```
# calculates the value of a set of regression evaluation metrics
regr.eval(algae[, "a1"], rt.predictions.a1, train.y = algae[,
                                                            "a1"])
```

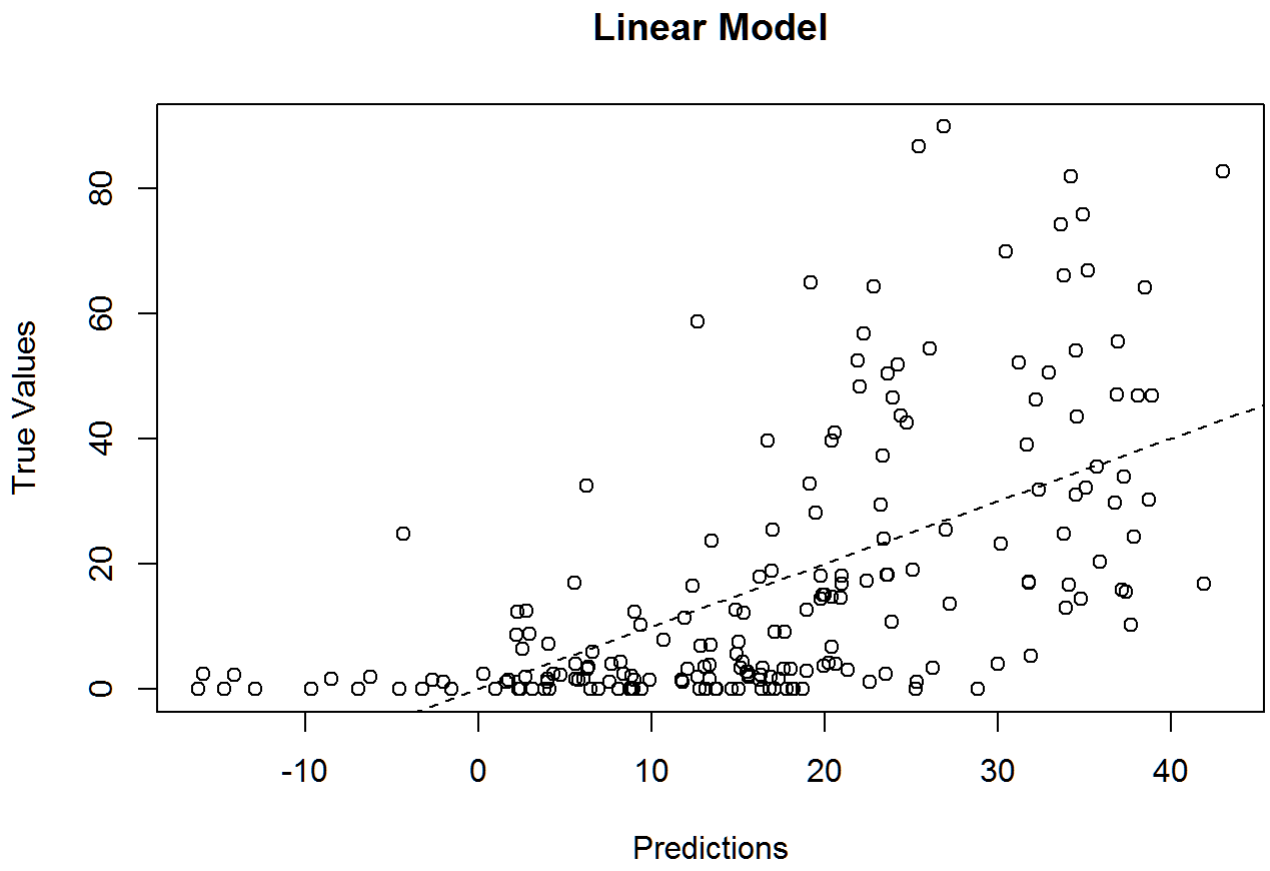
```
##         mae         mse        rmse        mape        nmse        nmae
##  10.3624227 227.0338940  15.0676439         Inf   0.4972574   0.6202654
```

```
#uses scatter plots of the errors
old.par <- par(mfrow = c(1, 2))
plot(lm.predictions.a1, algae[, "a1"], main = "Linear Model",
     xlab = "Predictions", ylab = "True Values")
abline(0, 1, lty = 2)
plot(rt.predictions.a1, algae[, "a1"], main = "Regression Tree",
     xlab = "Predictions", ylab = "True Values")
abline(0, 1, lty = 2)
```






```
par(old.par)

#this model predicts negative algae frequencies
plot(lm.predictions.a1,algae[,'a1'],main="Linear Model",
     xlab="Predictions",ylab="True Values")
abline(0,1,lty=2)
#here we see the rows of the algae data frame corresponding to the clicked circles
algae[identify(lm.predictions.a1,algae[,'a1']),]
```




```
##  [1] season size   speed  mxPH   mnO2   Cl     NO3    NH4    oPO4   PO4
## [11] Chla   a1     a2     a3     a4     a5     a6     a7
## <0 rows> (or 0-length row.names)
```

```
sensible.lm.predictions.a1 <- ifelse(lm.predictions.a1 <
                                     0, 0, lm.predictions.a1)
regr.eval(algae[, "a1"], lm.predictions.a1, stats = c("mae",
                                                      "mse"))
```

```
##       mae       mse
##  13.10681 295.54069
```

```
regr.eval(algae[, "a1"], sensible.lm.predictions.a1, stats = c("mae",
                                                               "mse"))
```

```
##       mae       mse
##  12.48276 286.28541
```

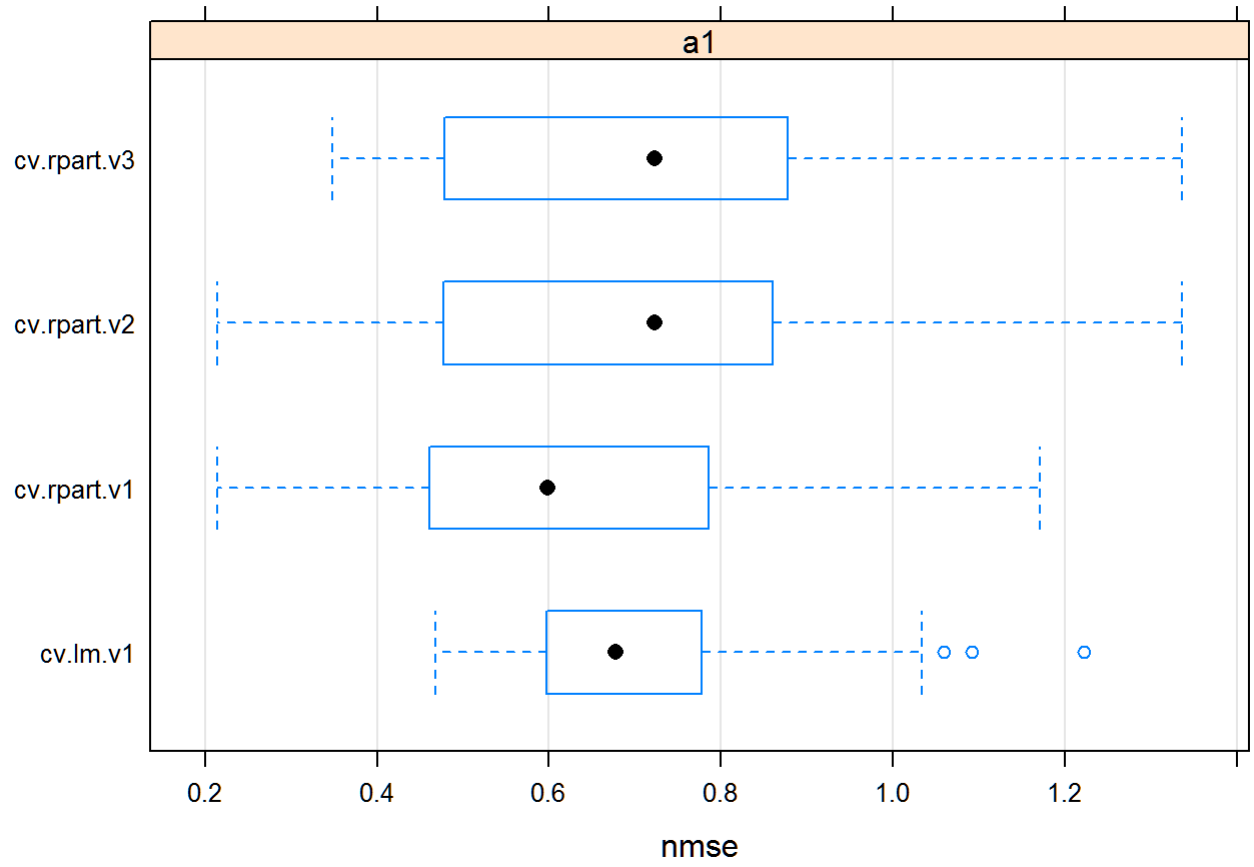
```
cv.rpart <- function(form,train,test,...) {
  m <- rpartXse(form,train,...)
  p <- predict(m,test)
  mse <- mean((p-resp(form,test))^2)
  c(nmse=mse/mean((mean(resp(form,train))-resp(form,test))^2))
}
cv.lm <- function(form,train,test,...) {
  m <- lm(form,train,...)
  p <- predict(m,test)
  p <- ifelse(p < 0,0,p)
  mse <- mean((p-resp(form,test))^2)
  c(nmse=mse/mean((mean(resp(form,train))-resp(form,test))^2))
}

res <- experimentalComparison(
  c(dataset(a1 ~ .,clean.algae[,1:12],'a1')),
  c(variants('cv.lm'),
    variants('cv.rpart',se=c(0,0.5,1))),
  cvSettings(3,10,1234))
```

```
##
##
## #####  CROSS VALIDATION  EXPERIMENTAL COMPARISON #####
##
## ** DATASET :: a1
##
## ++ LEARNER :: cv.lm  variant ->  cv.lm.v1
##
##  3 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v1
##
##  3 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v2
##
##  3 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v3
##
##  3 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
```

```
#prints the summary of the cross-validation experimentation results
summary(res)
```

```
##
## == Summary of a  Cross Validation  Experiment ==
##
##  3 x 10 - Fold Cross Validation run with seed =  1234
##
## * Data sets ::  a1
## * Learners  ::  cv.lm.v1, cv.rpart.v1, cv.rpart.v2, cv.rpart.v3
##
## * Summary of Experiment Results:
##
##
## -> Datataset:  a1
##
##  *Learner: cv.lm.v1
##              nmse
## avg     0.7196105
## std     0.1833064
## min     0.4678248
## max     1.2218455
## invalid 0.0000000
##
##  *Learner: cv.rpart.v1
##              nmse
## avg     0.6440843
## std     0.2521952
## min     0.2146359
## max     1.1712674
## invalid 0.0000000
##
##  *Learner: cv.rpart.v2
##              nmse
## avg     0.6873747
## std     0.2669942
## min     0.2146359
## max     1.3356744
## invalid 0.0000000
##
##  *Learner: cv.rpart.v3
##              nmse
## avg     0.7167122
## std     0.2579089
## min     0.3476446
## max     1.3356744
## invalid 0.0000000
```

```
#plots the visualisation of the above results
plot(res)
```




```
#we can  know the specific parameter settings corresponding to any label using 'getVariant'
getVariant("cv.rpart.v1", res)
```

```
##
## Learner::  "cv.rpart"
##
## Parameter values
##   se  =  0
```

```
#to perform comparative experiment for all seven prediction tasks we are facing at the same time
DSs <- sapply(names(clean.algae)[12:18],
              function(x,names.attrs) {
                f <- as.formula(paste(x,"~ ."))
                dataset(f,clean.algae[,c(names.attrs,x)],x)
              },
              names(clean.algae)[1:11])
res.all <- experimentalComparison(
  DSs,
  c(variants('cv.lm'),
    variants('cv.rpart',se=c(0,0.5,1))
  ),
  cvSettings(5,10,1234))
```

```
##
##
## #####  CROSS VALIDATION  EXPERIMENTAL COMPARISON #####
##
## ** DATASET :: a1
##
## ++ LEARNER :: cv.lm  variant ->  cv.lm.v1
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v1
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v2
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v3
##
```

```
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ** DATASET :: a2
##
## ++ LEARNER :: cv.lm  variant ->  cv.lm.v1
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v1
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v2
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
```

```
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v3
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ** DATASET :: a3
##
## ++ LEARNER :: cv.lm  variant ->  cv.lm.v1
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v1
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v2
```

```
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v3
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ** DATASET :: a4
##
## ++ LEARNER :: cv.lm  variant ->  cv.lm.v1
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v1
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
```

```
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v2
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v3
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ** DATASET :: a5
##
## ++ LEARNER :: cv.lm  variant ->  cv.lm.v1
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
```

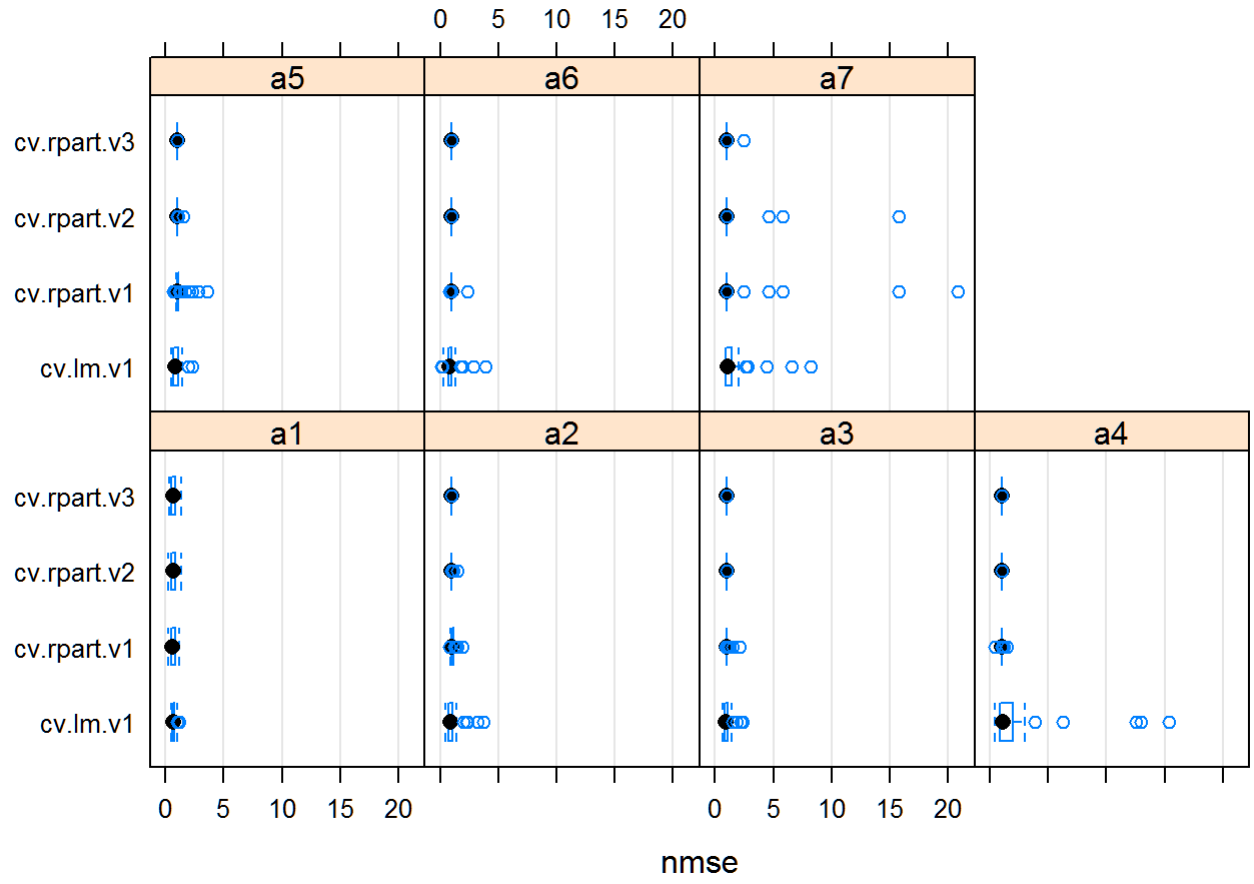
```
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v1
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v2
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v3
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ** DATASET :: a6
##
## ++ LEARNER :: cv.lm  variant ->  cv.lm.v1
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
```

```
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v1
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v2
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v3
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ** DATASET :: a7
```

```
##
## ++ LEARNER :: cv.lm  variant ->  cv.lm.v1
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v1
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v2
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v3
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
```

```
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
```

```
#plots the results of the models for the di???erent algae on the CV process.
plot(res.all)
```




```
#checks which is the best model for each problem
bestScores(res.all)
```

```
## $a1
##           system   score
## nmse cv.rpart.v1 0.64231
##
## $a2
##           system score
## nmse cv.rpart.v3     1
##
## $a3
##           system score
## nmse cv.rpart.v2     1
##
## $a4
##           system score
## nmse cv.rpart.v2     1
##
## $a5
##        system     score
## nmse cv.lm.v1 0.9316803
##
## $a6
##        system     score
## nmse cv.lm.v1 0.9359697
##
## $a7
##           system    score
## nmse cv.rpart.v3 1.029505
```

```
#Random Forest
#install.packages("randomForest")
#Loads the package "randomForest"
library(randomForest)
```

```
## randomForest 4.6-14
```

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

```
##
## Attaching package: 'randomForest'
```

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

```
cv.rf <- function(form,train,test,...) {
  m <- randomForest(form,train,...)
  p <- predict(m,test)
  mse <- mean((p-resp(form,test))^2)
  c(nmse=mse/mean((mean(resp(form,train))-resp(form,test))^2))
}

#to carry out the cross validation comparison
res.all <- experimentalComparison(
  DSs,
  c(variants('cv.lm'),
    variants('cv.rpart',se=c(0,0.5,1)),
    variants('cv.rf',ntree=c(200,500,700))
  ),
  cvSettings(5,10,1234))
```

```
##
##
## #####  CROSS VALIDATION  EXPERIMENTAL COMPARISON #####
##
## ** DATASET :: a1
##
## ++ LEARNER :: cv.lm  variant ->  cv.lm.v1
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v1
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v2
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v3
##
```

```
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rf  variant ->  cv.rf.v1
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rf  variant ->  cv.rf.v2
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rf  variant ->  cv.rf.v3
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
```

```
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ** DATASET :: a2
##
## ++ LEARNER :: cv.lm  variant ->  cv.lm.v1
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v1
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v2
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v3
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
```

```
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rf  variant ->  cv.rf.v1
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rf  variant ->  cv.rf.v2
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rf  variant ->  cv.rf.v3
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
```

```
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ** DATASET :: a3
##
## ++ LEARNER :: cv.lm  variant ->  cv.lm.v1
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v1
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v2
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v3
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
```

```
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rf  variant ->  cv.rf.v1
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rf  variant ->  cv.rf.v2
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rf  variant ->  cv.rf.v3
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
```

```
##
##
## ** DATASET :: a4
##
## ++ LEARNER :: cv.lm  variant ->  cv.lm.v1
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v1
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v2
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v3
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
```

```
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rf  variant ->  cv.rf.v1
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rf  variant ->  cv.rf.v2
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rf  variant ->  cv.rf.v3
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
```

```
##
## ** DATASET :: a5
##
## ++ LEARNER :: cv.lm  variant ->  cv.lm.v1
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v1
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v2
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v3
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
```

```
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rf  variant ->  cv.rf.v1
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rf  variant ->  cv.rf.v2
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rf  variant ->  cv.rf.v3
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
```

```
## ** DATASET :: a6
##
## ++ LEARNER :: cv.lm  variant ->  cv.lm.v1
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v1
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v2
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v3
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
```

```
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rf  variant ->  cv.rf.v1
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rf  variant ->  cv.rf.v2
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rf  variant ->  cv.rf.v3
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ** DATASET :: a7
```

```
##
## ++ LEARNER :: cv.lm  variant ->  cv.lm.v1
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v1
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v2
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rpart  variant ->  cv.rpart.v3
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
```

```
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rf  variant ->  cv.rf.v1
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rf  variant ->  cv.rf.v2
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
##
##
## ++ LEARNER :: cv.rf  variant ->  cv.rf.v3
##
##  5 x 10 - Fold Cross Validation run with seed =  1234
## Repetition  1
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  2
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  3
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  4
## Fold:  1  2  3  4  5  6  7  8  9  10
## Repetition  5
## Fold:  1  2  3  4  5  6  7  8  9  10
```

```
#confirms the advantages of the ensemble approach
bestScores(res.all)
```

```
## $a1
##        system     score
## nmse cv.rf.v3 0.5467636
##
## $a2
##        system     score
## nmse cv.rf.v3 0.7695782
##
## $a3
##           system score
## nmse cv.rpart.v2     1
##
## $a4
##        system     score
## nmse cv.rf.v1 0.9728596
##
## $a5
##        system     score
## nmse cv.rf.v2 0.7916332
##
## $a6
##        system    score
## nmse cv.rf.v2 0.911758
##
## $a7
##           system    score
## nmse cv.rpart.v3 1.029505
```

```
#Statistical Significance Analysis of Comparison Results
compAnalysis(res.all,against='cv.rf.v3',
             datasets=c('a1','a2','a4','a6'))
```

```
##
## == Statistical Significance Analysis of Comparison Results ==
##
## Baseline Learner::    cv.rf.v3  (Learn.1)
##
## ** Evaluation Metric::    nmse
##
## - Dataset: a1
##       Learn.1   Learn.2 sig.2   Learn.3 sig.3   Learn.4 sig.4   Learn.5
## AVG 0.5467636 0.7077282    ++ 0.6423100    +  0.6569726    ++ 0.6875212
## STD 0.1727235 0.1639373       0.2399321       0.2397636       0.2348946
##     sig.5   Learn.6 sig.6   Learn.7 sig.7
## AVG    ++ 0.5505008       0.5473338
## STD       0.1783960       0.1724374
##
## - Dataset: a2
##       Learn.1   Learn.2 sig.2   Learn.3 sig.3    Learn.4 sig.4
## AVG 0.7695782 1.0449317    ++ 1.0426327    ++ 1.01626123    ++
## STD 0.1431761 0.6276144       0.2005522       0.07435826
##          Learn.5 sig.5   Learn.6 sig.6   Learn.7 sig.7
## AVG 1.000000e+00    ++ 0.7775628       0.7744307
## STD 2.389599e-16       0.1473327       0.1462083
##
## - Dataset: a4
##       Learn.1  Learn.2 sig.2   Learn.3 sig.3      Learn.4 sig.4
## AVG 0.9746980 2.111976       1.0073953    +  1.000000e+00    +
## STD 0.3823094 3.118196       0.1065607       2.774424e-16
##          Learn.5 sig.5   Learn.6 sig.6   Learn.7 sig.7
## AVG 1.000000e+00    +  0.9728596       0.9833417
## STD 2.774424e-16       0.3515190       0.3829643
##
## - Dataset: a6
##       Learn.1   Learn.2 sig.2   Learn.3 sig.3      Learn.4 sig.4
## AVG 0.9133912 0.9359697    ++ 1.0191041       1.000000e+00
## STD 0.3573499 0.6045963       0.1991436       2.451947e-16
##          Learn.5 sig.5   Learn.6 sig.6   Learn.7 sig.7
## AVG 1.000000e+00       0.9275673       0.9117580
## STD 2.451947e-16       0.3793325       0.3757454
##
## Legends:
## Learners -> Learn.1 = cv.rf.v3 ; Learn.2 = cv.lm.v1 ; Learn.3 = cv.rpart.v1 ; Learn.4 = cv.rp
art.v2 ; Learn.5 = cv.rpart.v3 ; Learn.6 = cv.rf.v1 ; Learn.7 = cv.rf.v2 ;
## Signif. Codes -> 0 '++' or '--' 0.001 '+' or '-' 0.05 ' ' 1
```

```
#to obtain all seven models
bestModelsNames <- sapply(bestScores(res.all),
                          function(x) x['nmse','system'])
learners <- c(rf='randomForest',rpart='rpartXse')
funcs <- learners[sapply(strsplit(bestModelsNames,'\\.'),
                         function(x) x[2])]
#gives the model corresponding to the variant name
parSetts <- lapply(bestModelsNames,
                   function(x) getVariant(x,res.all)@pars)
bestModels <- list()
#for(a in 1:7) {
#  form <- as.formula(paste(names(clean.algae)[11+a],'~ .'))
#  bestModels[[a]] <- do.call(funcs[a],
#                             c(list(form,clean.algae[,c(1:11,11+a)]),parSetts[[a]]))
#}

#fills unknowns on a test set
clean.test.algae <- knnImputation(test.algae, k = 10, distData = algae[,
                                                                       1:11])

#prints the matrix with the predictions for the entire test set
preds <- matrix(ncol=7,nrow=140)
#for(i in 1:nrow(clean.test.algae))
#  preds[i,] <- sapply(1:7,
#                      function(x)
#                        predict(bestModels[[x]],clean.test.algae[i,])
#  )

#calculates the NMSE scores of our models
avg.preds <- apply(algae[,12:18],2,mean)
apply( ((algae.sols-preds)^2), 2,mean) /
  apply( (scale(algae.sols,avg.preds,F)^2),2,mean)
```

```
## a1 a2 a3 a4 a5 a6 a7
## NA NA NA NA NA NA NA
```