Customizing the rpart library for multivariate gaussian outcomes: the longRPart library

Sam Stewart[1], MMath, AStat
Mohammed Abdolell[2], MSc, PStat
Michael Leblanc[3], PhD

[1] IDPhD Candidate (Health Informatics), Faculty of Comp Sci
Statistical Consulting Service, Faculty of Math and Stats
Dalhousie University
[2] Dept of Diagnostic Radiology and Division of Medical Education, Dalhousie University
Dalla Lana School of Public Health, University of Toronto
[3] Fred Hutchinson Cancer Research Center

- In a paper written in 2002, Abdolell et al. [1] outlined a method for creating classification trees for multivariate normal data.
- This project implemented and extended this algorithm in an R package.
- The outline of the presentation:
  - Problem description
  - Solution in R
  - Example data

Problem

- Consider the situation where we have n patients with p observations per patient, along with a vector of patient attributes.

$$\mathbf{Y}_{n\times p}, \mathbf{X}_{n\times 1}$$

- Assume that the outcomes are *multivariate normal*, that is

$$f(\mathbf{y}_i; \mu_i, \Sigma) = (2\pi)^{-p/2}|\Sigma|^{-1/2} exp\left(-\frac{1}{2}\left[(y_i - \mu_i)'\Sigma^{-1}(y_i - \mu_i)\right]\right)$$

- Abdolell et. al [1] proposed a method for building classification trees for longitudinal data, based on likelihood ratio statistics

Building Classification Trees

- Using maximum likelihood estimates we get $\mu_i = \mathbf{y}_i$, and the maximum log-likelihood fuction

$$l(y_i; y_i) = -\frac{1}{2} log\left[(2\pi)^p |\Sigma|\right] \quad (1)$$

- The deviance function for a single observation, therefore, is

$$D(\mu_i; y_i) = l(y_i; y_i) - l(\mu_i; y_i) = \ldots = (y_i - \mu_i)' \Sigma^{-1} (y_i - \mu_i) \quad (2)$$

- The deviance function is the **Mahalanobis distance** [2] for the observations. In practice this can be obtained from a mixed model of the data (more on this later)

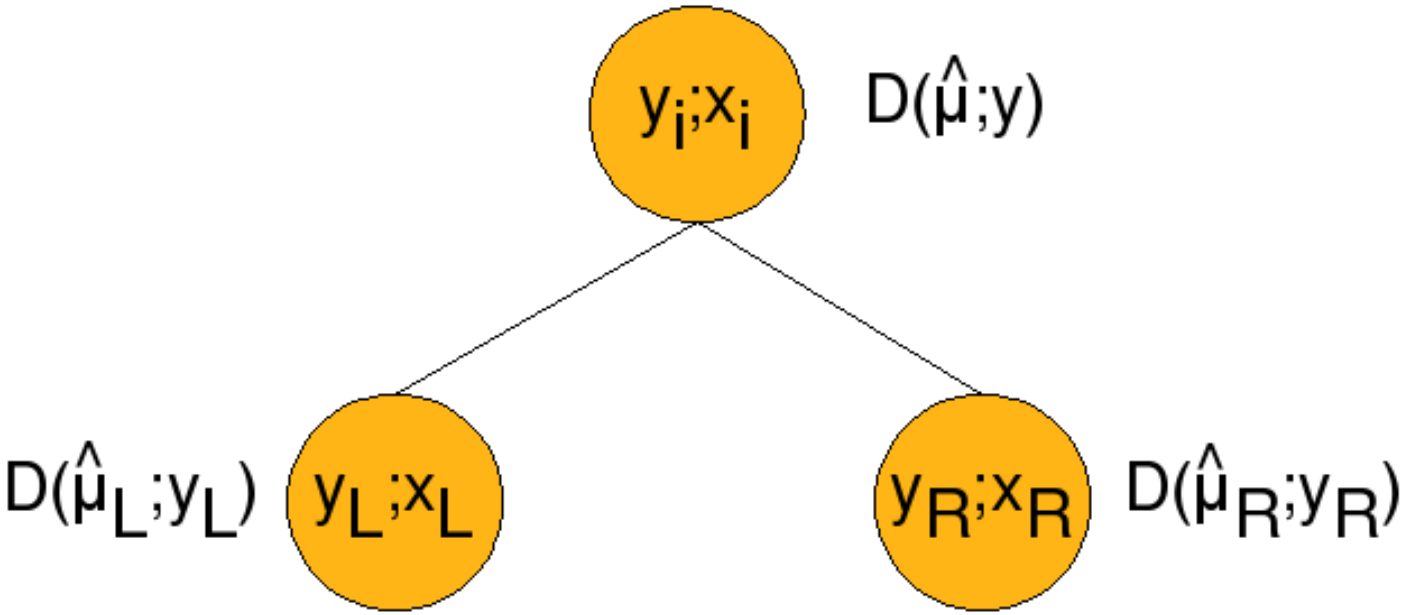
- The deviance for a set of observations will be the sum of deviances for those observations

$$D(\hat{\mu}; y) = \sum_{i=1}^{n} D(\hat{\mu}_i; y_i) \quad (3)$$

Partitioning

- The partitioning step is the same for multivariate outcomes is the same as it is for univariate outcomes.
- The data is partitioned by the values of the patient attribute ($\mathbf{X}_{n\times 1}$) and the deviance is calculated for each child node. The optimal split is the one that creates the greatest reduction in deviance between the parent node and the two child nodes.
- let s be a particular split in the set of all possible splits S, and let N be the dataset, split into N_R, N_L

Partitioning



We then define the splitting criterion as

$$\Delta D(s,N) = D_N(\hat{\mu};,y) - D_{N_L,N_R}(\hat{\mu}_L,\hat{\mu}_R;y) \quad (4)$$

$$= \sum_{i=1}^{n} D(\hat{\mu};y_i) - \left[\sum_{i=1}^{n_L} D(\hat{\mu};y_i) + \sum_{i=1}^{n_R} D(\hat{\mu};y_i)\right] \quad (5)$$

Partitioning

- The best split, therefore, is the split that maximizes $\Delta D(s, N)$
- Along with this partitioning function, Abdolell et al. provide the theory for a permutation test to obtain a p-value and a bootstrapping function to obtain a confidence interval on the values of the first split.
- Though not covered in the original paper, the extension to multiple splits and multiple patient attributes is implied
- As well v-fold cross-validation can be implemented

Solution in R

Providing the algorithm

- The original paper explained the implementation of the algorithm in SAS using the MIXED procedure.
- The algorithm has since been adapted to R, and is available from CRAN as the package *longRPart*
- The longRPart library has added to the partitioning function by providing plotting functions, multiple splits and predictors, and cross-validation.

longRPart

- The library is heavily dependant on two other R libraries:

  `rpart` is the traditional method of creating classification and regression trees.

  `nlm` provides the methods for fitting the data and obtaining the Mahalanobis distance.

- The library uses the algorithms in `rpart` to extend its partitioning to multivariate normal outcomes.

Customizing rpart

- Customizing *rpart* requires three functions: **evaluation**, **split** and **init**.
- These three functions are passed as a list to the *rpart* function through the `method` option.

```
rpart(y∼x, data=dat, method=list(eval=evaluation,
split=split, init=initialize))
```

evaluation

- The evaluation function evaluates the deviance of a node.
- The function is called once *per node*.
- The function returns a list of two variables: a label for the node and a measure of deviance at the node
- For the *longRPart* library, the label at the node is the estimate of $\hat{\mu}$ for the observations at the node, and the deviance is calculated as the $-2\times$**log-likelihood** of the model.

evaluation

```
evaluation <- function(y, wt, parms){
  model = lme(lmeFormula,data=parms[groupingFactor%in%y,]
    if(continuous){
      slope = model$coefficients$fixed[2]
    }
    else{
      levels = length(levels(data[,names(data)==terms[1]]
      y=model$coefficients$fixed[1:levels]
      x=1:levels
      slope = lm(y~x)$coefficients[2]
    }
    list(label=slope,deviance=-2*(model$logLik))
  }
```

split

- The function is called once per node *per covariate*.
- The `split` function evaluates the potential splitting values to determine what the best split is.
- The idea is to check every candidate value for a covariate and provide an ordered list of the results.
- This function is very computationally intensive in *longRPart*: For every potential split value two mixed models need to be fit, one for the children on the left and one for the children on the right.

split

- The function returns two vectors:

  goodness is a measure of the split, with larger values being better, and a value of 0 indicating no split. This vector is $\Delta D(s, N)$.

  direction is the applied ordering for categorical data, or a vector of -1/1 indicating in which direction the split should be directed for continuous data.

split

```
split <- function(){
  calculate root deviance
      if(categorical){
          establish the ordering of the categorical variables
      }
      for(i in xUnique){
        1.  partition data by observation i
        2.  calculate deviance for each dataset
        3.  Save the sum of the child deviances in a vector
      }
      return (goodness=rootDev - node deviance, direction=dir)
}
```

init

- Is used to initialize the process
- Allows the passing of auxiliary variables
- Includes two functions: *summary* helps summarize the model in text, while *text* to add text to the plot of the tree.

init

```
initialize <- function(y,offset,parms=0,wt){
  list(
  y=y,
  parms=parms,
  numresp=1,
  numy=1,
  summary=function(yval,dev,wt,ylevel,digits){
    paste("deviance (-2logLik)",
      format(signif(dev),3),
      "slope",
      signif(yval,2))
  },
  text= function(yval,dev,wt,ylevel,digits,n,use.n){
    if(!use.n){paste("m:",format(signif(yval,1)))}
    else{paste("n:",n)}
  }
  )
}
```

Other functions

Working with longRPart models

lrpPVal This function uses permutations of the covariates to determine the p-value of the initial split

lrpCI This function takes bootstrap samples of the data to calculates a confidence interval for the initial split

lrpCV This function performs v-fold cross validation of the model to assess its fit

lrpPlot, lrpTreePlot These are the two plotting functions for the data

Application

Data

- Data is on cochlear implants
- Question is the effectiveness of the cochlear implants on improving speech and language skills
- Independant variable of interest is age: is the effect of the implant varried by age.
- Dependent variable of interest is speech perception scores taken repeatedly over time.

Analysis

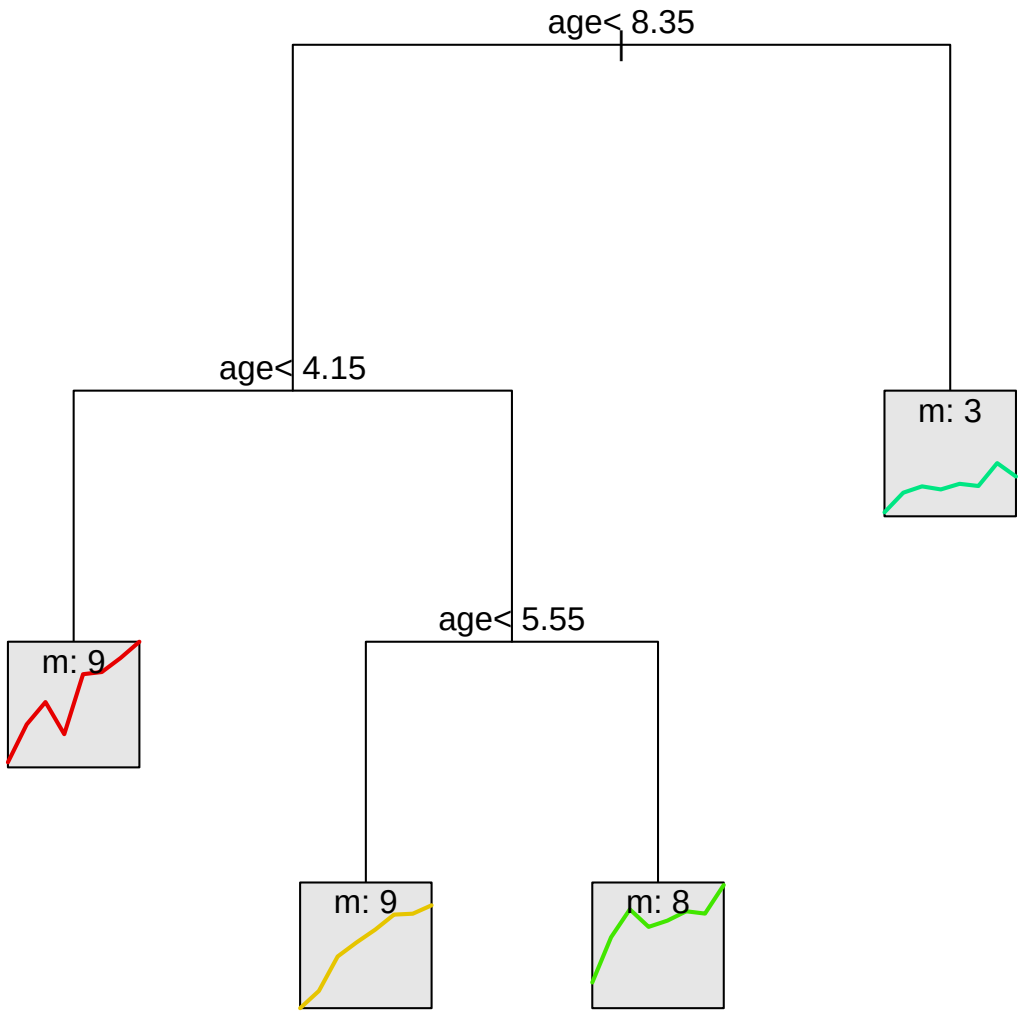
```
data(pbkphData)
model = longRPart(pbkph~Time, ~age+gender,
~1|Subject, pbkphData, R=corExp(form=~time),
control=rpart.control(minbucket=80,xval=10))
```

Model Summary

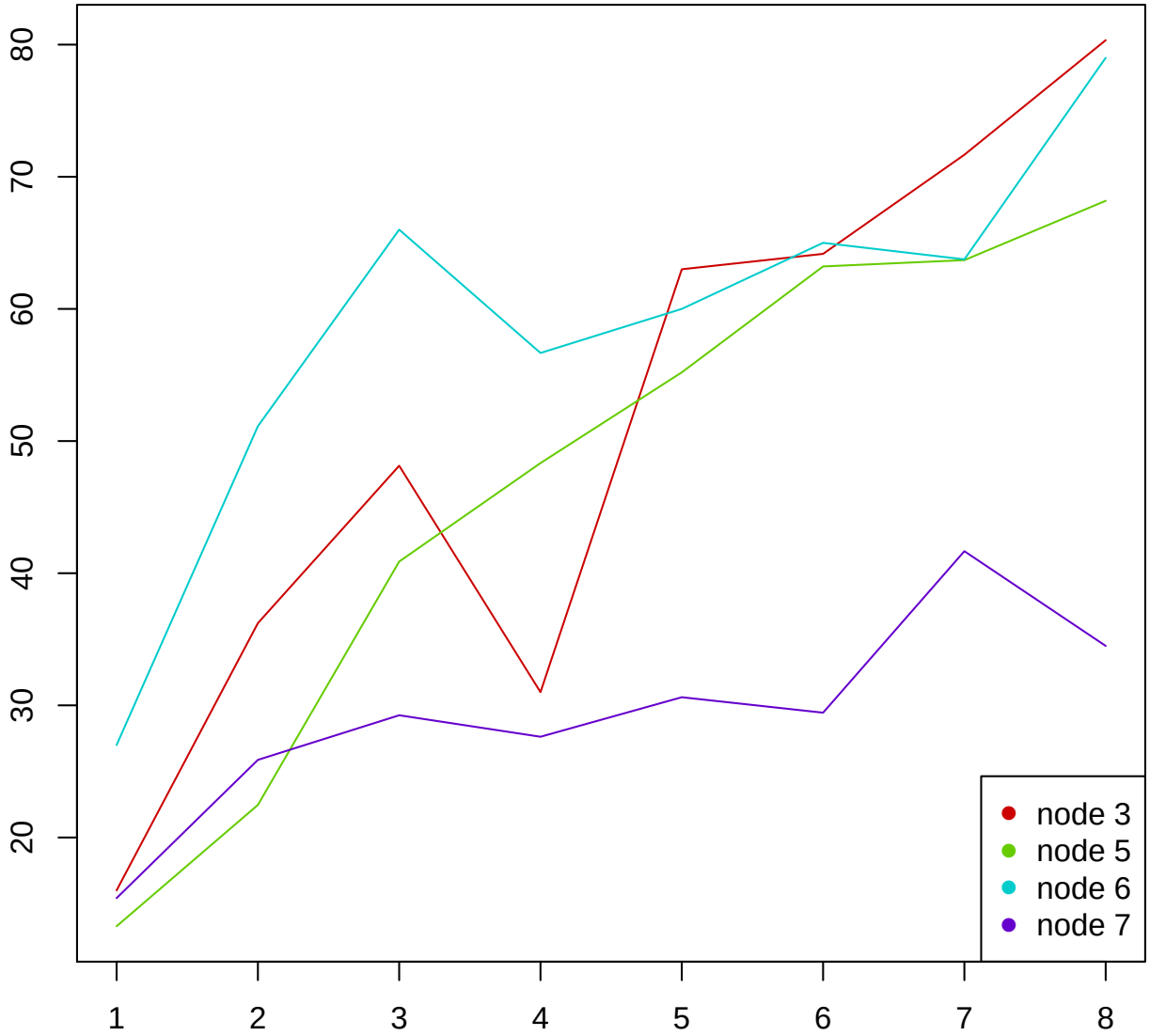
summary(model)

```
Call:
rpart(formula = paste(groupingName, c(rPartFormula)), data = data,
    method = list(eval = evaluation, split = split, init = initiali
    parms = data, control = control)
  n= 552

          CP nsplit rel error
1 0.04025716      0 1.0000000
2 0.02922562      1 0.9597428
3 0.02802864      2 0.9305172
4 0.01000000      3 0.9024886

Node number 1: 552 observations,    complexity param=0.04025716
deviance (-2logLik) 2205.360 slope 5
  left son=2 (416 obs) right son=3 (136 obs)
  Primary splits:
      age    < 8.35 to the left,  improve=1147.072, (0 missing)
      gender splits as  LR,       improve=1124.425, (0 missing)
```

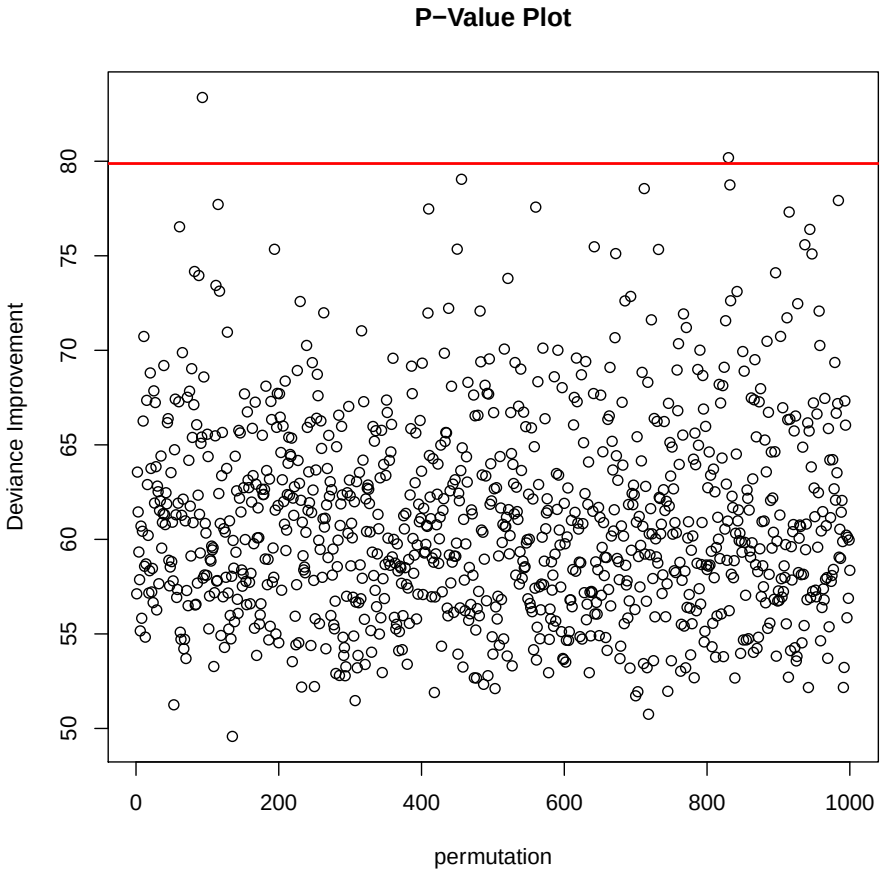
lrpTreePlot

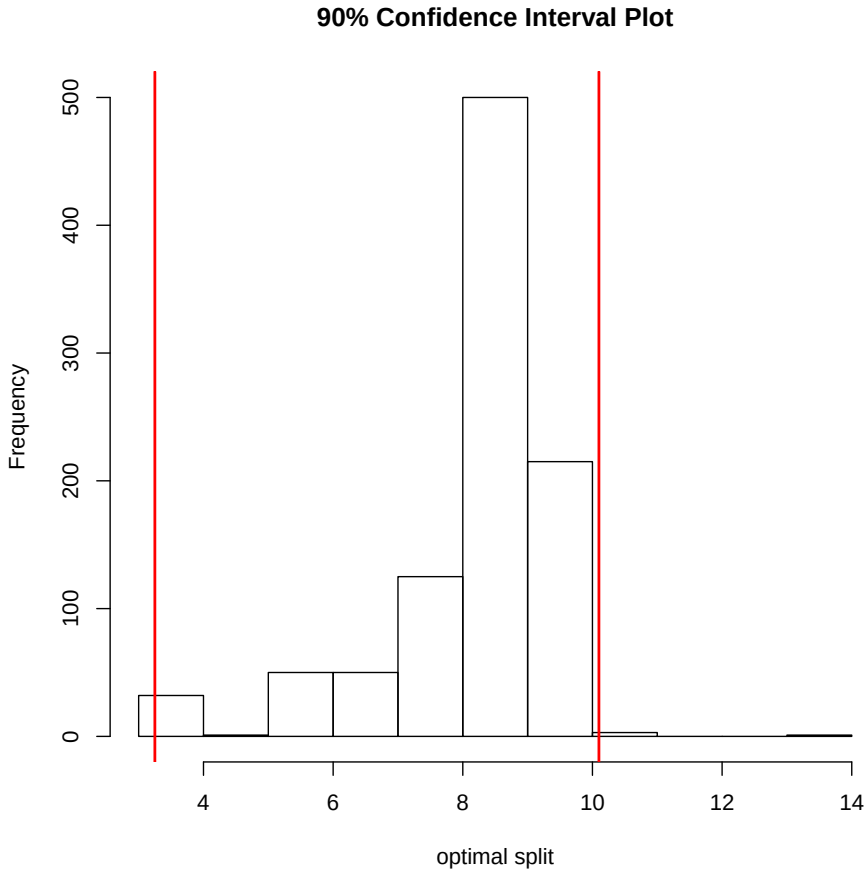


lrpPlot

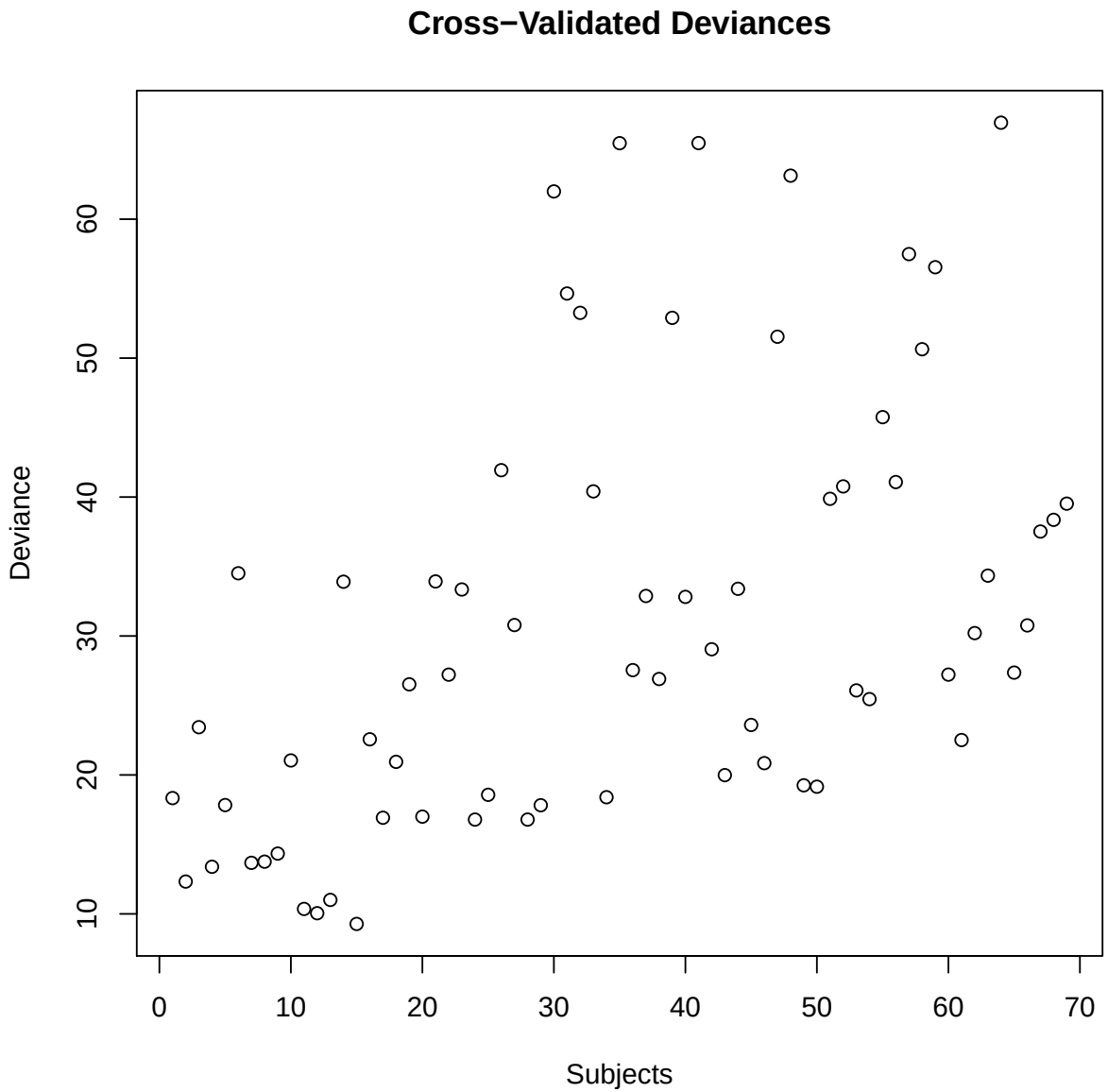


P-Values and Confidence Intervals





CV


Cross−Validated Deviances
Deviance
Subjects
10
20
30
40
50
60
0
10
20
30
40
50
60
70

Conclusions

- Additional work is required to address some issues
  - Cross-Validation
  - S3 Compatibility
  - Improving the speed of the Mahlanobis computation (any ideas?)
- It is possible to use the existing algorithms in R to extend classification trees to non-traditional outcomes.

Thank you

Merci Beaucoup!


M. Abdolell, M. LeBlanc, D. Stephens, and R. V. Harrison.
Binary partitioning for continuous longitudinal data: categorizing a prognostic variable.
Statistics in Medicine, 21:3395–2409, 2002.


BFJ Manley.
Multivariate Statistical Methods: A Primer.
Chapman and Hall: London, 1986.