

# Decision Tree Week 5

Dan Burke

11/9/2021

```
#load needed libraries
```

```
library(stats)
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.1.1
```

```
##
```

```
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
```

```
##
```

```
##      filter, lag
```

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

```
##
```

```
##      intersect, setdiff, setequal, union
```

```
library(ggplot2)
```

```
library(ggfortify)
```

```
## Warning: package 'ggfortify' was built under R version 4.1.1
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.1.1
```

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

```
library(rpart)
```

```
library(rattle)
```

```
## Warning: package 'rattle' was built under R version 4.1.1
```

```
## Loading required package: tibble
```

```
## Loading required package: bitops
```

```
## Rattle: A free graphical interface for data science with R.
```

```
## Version 5.4.0 Copyright (c) 2006-2020 Togaware Pty Ltd.
```

```
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
#Data Preparation
```

```
fedpapers <- read.csv("C:\\Users\\danbu\\Desktop\\Applied Machine Learning\\Week 4\\HW4\\fedPapers85.csv")
```

```
#check for Complete Cases
```

```
sum(!complete.cases(fedpapers))
```

```
## [1] 0
```

```
#Find Unique Values for Authors
```

```
authors <- unique(fedpapers$author)
```

```
authors
```

```
## [1] "dispt"      "Hamilton" "HM"        "Jay"        "Madison"
```

```
#drop the file name column
```

```
fedpapers <- fedpapers[, !names(fedpapers) %in% c("filename")]
```

```
fedpapers <- fedpapers[fedpapers$author == "Hamilton" | fedpapers$author == "Madison" | fedpapers$author == "dispt",]
```

```
knownpapers <- fedpapers[fedpapers$author == "Hamilton" | fedpapers$author == "Madison" | fedpapers$author == "dispt",]
```

```
unknownpapers <- fedpapers[which(fedpapers$author == "HM" | fedpapers$author == "dispt"),]
```

```
#Create Train/Test Datasets, utilizing a 80/20 Split
```

```
create_train_test <- function(data, size = 0.8, train = TRUE) {
```

```
  n_row = nrow(data)
```

```
  total_row = size * n_row
```

```
  train_sample <- 1: total_row
```

```
  if (train == TRUE) {
```

```
    return (data[train_sample, ])
```

```
  } else {
```

```
    return (data[-train_sample, ])
```

```
  }
```

```
}
```

```
fedpapers_clean <- data.frame(fedpapers)
```

```
data_train <- fedpapers[fedpapers$author == "Hamilton" | fedpapers$author == "Madison" | fedpapers$author == "dispt",]
```

```
data_test <- fedpapers[which(fedpapers$author == "HM" | fedpapers$author == "dispt"),]
```

```
set.seed(2021)
```

```
sum(!complete.cases(data_train))
```

```
## [1] 0
```

```
sum(!complete.cases(data_test))
```

```
## [1] 0
```

```
data_train <- droplevels(data_train)
data_test <- droplevels(data_test)
```

```
dim(data_train)
```

```
## [1] 71 71
```

```
dim(data_test)
```

```
## [1] 11 71
```

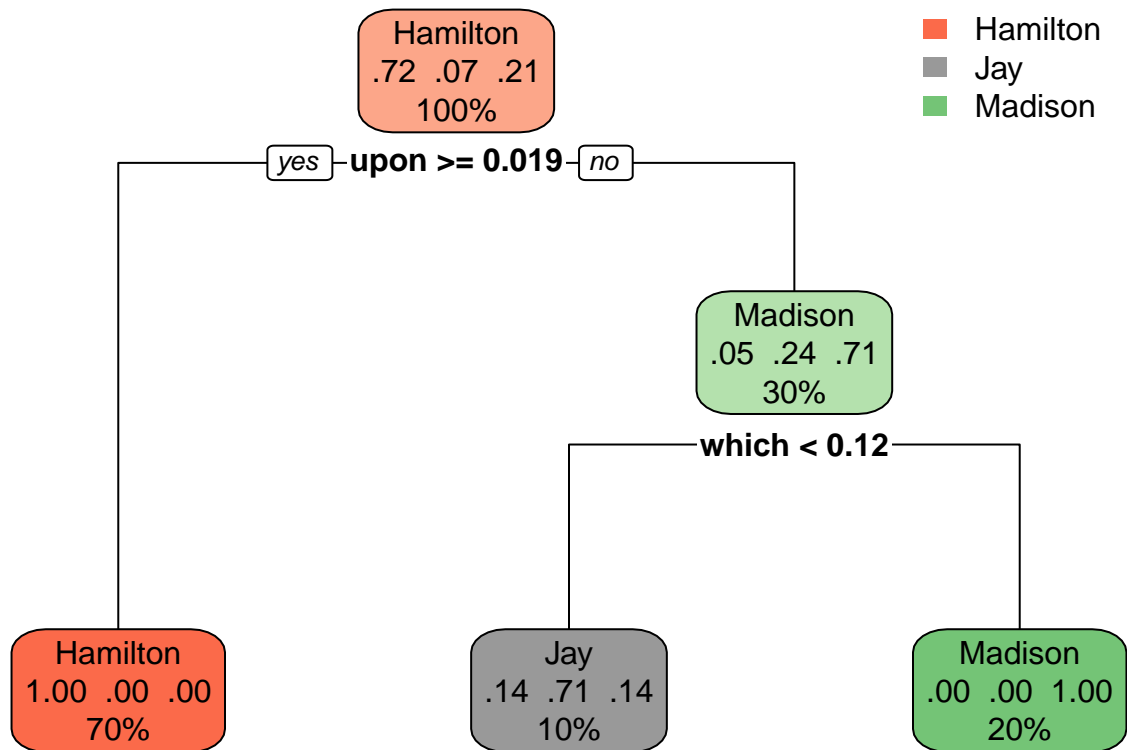
```
papers_model <- train(author ~ ., data= data_train, metric="Accuracy", method="rpart")
print(papers_model)
```

```
## CART
##
## 71 samples
## 70 predictors
## 3 classes: 'Hamilton', 'Jay', 'Madison'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 71, 71, 71, 71, 71, 71, ...
## Resampling results across tuning parameters:
##
##   cp   Accuracy   Kappa
##   0.0  0.9146575  0.8147111
##   0.2  0.9041654  0.7880220
##   0.7  0.7852914  0.3527466
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was cp = 0.
```

```
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 4.1.1
```

```
rpart.plot(papers_model$finalModel)
```



```
papers_predict<- predict(papers_model, newdata = data_test, na.action = na.omit, type = "prob")
head(papers_predict, 5)
```

```
##      Hamilton      Jay  Madison
## 1 0.0000000 0.0000000 1.0000000
## 2 0.1428571 0.7142857 0.1428571
## 3 0.1428571 0.7142857 0.1428571
## 4 0.0000000 0.0000000 1.0000000
## 5 0.0000000 0.0000000 1.0000000
```

```
print(papers_predict)
```

```
##      Hamilton      Jay  Madison
## 1 0.0000000 0.0000000 1.0000000
## 2 0.1428571 0.7142857 0.1428571
## 3 0.1428571 0.7142857 0.1428571
## 4 0.0000000 0.0000000 1.0000000
## 5 0.0000000 0.0000000 1.0000000
## 6 0.0000000 0.0000000 1.0000000
## 7 0.0000000 0.0000000 1.0000000
## 8 0.0000000 0.0000000 1.0000000
## 9 0.0000000 0.0000000 1.0000000
```

```
## 10 0.0000000 0.0000000 1.0000000
## 11 0.0000000 0.0000000 1.0000000
```

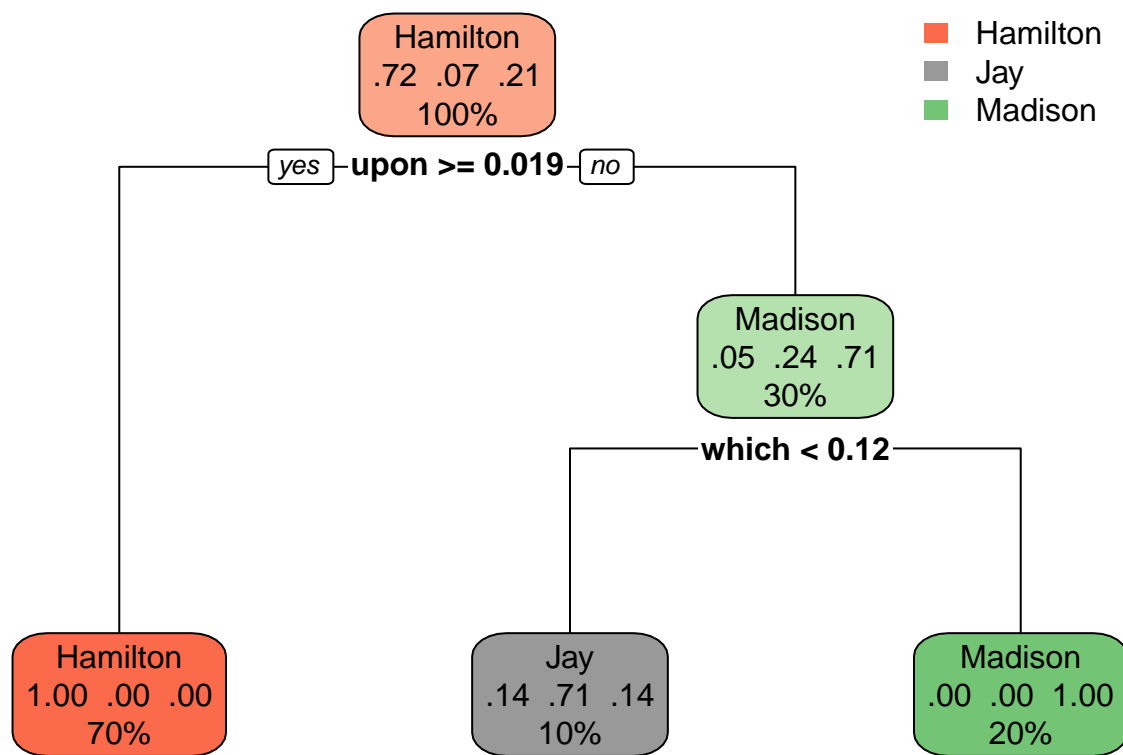
```
papers_predict_raw <- predict(papers_model, newdata = data_test, na.action = na.omit, type = "raw")
print(papers_predict_raw)
```

```
## [1] Madison Jay Jay Madison Madison Madison Madison Madison Madison
## [10] Madison Madison
## Levels: Hamilton Jay Madison
```

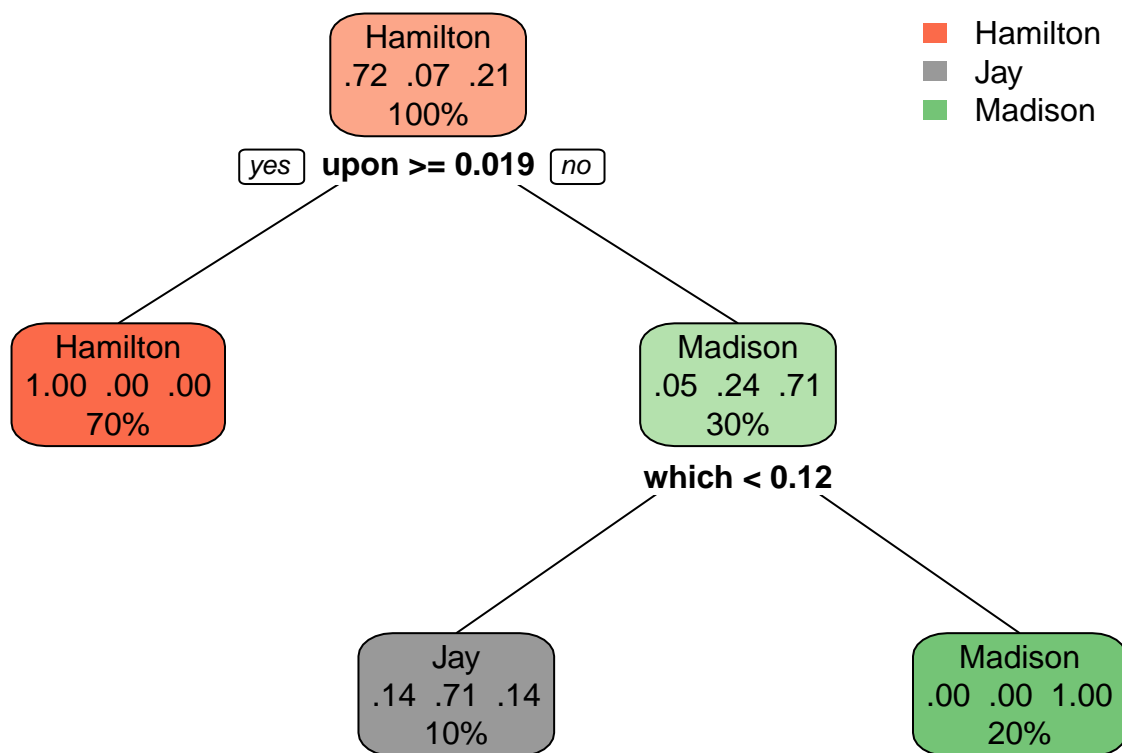
```
papers_model_prob_tuned <- train(author ~., data= data_train, method = "rpart",
                                metric = "Accuracy",
                                tuneLength= 9)
print(papers_model_prob_tuned$finalModel)
```

```
## n= 71
##
## node), split, n, loss, yval, (yprob)
##      * denotes terminal node
##
## 1) root 71 20 Hamilton (0.71830986 0.07042254 0.21126761)
##    2) upon>=0.019 50 0 Hamilton (1.00000000 0.00000000 0.00000000) *
##    3) upon< 0.019 21 6 Madison (0.04761905 0.23809524 0.71428571)
##      6) which< 0.115 7 2 Jay (0.14285714 0.71428571 0.14285714) *
##      7) which>=0.115 14 0 Madison (0.00000000 0.00000000 1.00000000) *
```

```
rpart.plot(papers_model_prob_tuned$finalModel)
```



```
rpart.plot(papers_model_prob_tuned$finalModel, fallen.leaves = FALSE)
```



```

predict_firstPass_tuned <- predict(papers_model_prob_tuned, newdata = data_test, na.action = na.omit, type = "prob")
print(predict_firstPass_tuned)

```

```

##      Hamilton      Jay    Madison
## 1  0.0000000 0.0000000 1.0000000
## 2  0.1428571 0.7142857 0.1428571
## 3  0.1428571 0.7142857 0.1428571
## 4  0.0000000 0.0000000 1.0000000
## 5  0.0000000 0.0000000 1.0000000
## 6  0.0000000 0.0000000 1.0000000
## 7  0.0000000 0.0000000 1.0000000
## 8  0.0000000 0.0000000 1.0000000
## 9  0.0000000 0.0000000 1.0000000
## 10 0.0000000 0.0000000 1.0000000
## 11 0.0000000 0.0000000 1.0000000

```

## Second Pass - Decision Tree, Omission of “Jay”

```

data_train <- fedpapers[fedpapers$author == "Hamilton" | fedpapers$author == "Madison",]
data_test <- fedpapers[which(fedpapers$author == "HM" | fedpapers$author == "dispt"),]

```

```
sum(!complete.cases(data_train))
```

```
## [1] 0
```

```
sum(!complete.cases(data_test))
```

```
## [1] 0
```

```
data_train <- droplevels(data_train)
```

```
data_test <- droplevels(data_test)
```

```
dim(data_train)
```

```
## [1] 66 71
```

```
dim(data_test)
```

```
## [1] 11 71
```

```
papers_model <- train(author ~ ., data= data_train, metric="Accuracy", method="rpart")  
print(papers_model$finalModel)
```

```
## n= 66
```

```
##
```

```
## node), split, n, loss, yval, (yprob)
```

```
##      * denotes terminal node
```

```
##
```

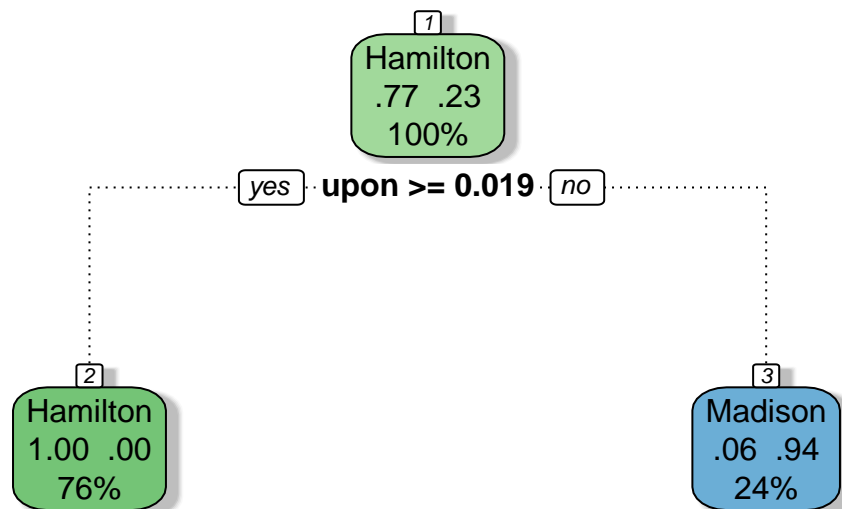
```
## 1) root 66 15 Hamilton (0.7727273 0.2272727)
```

```
##   2) upon>=0.019 50  0 Hamilton (1.0000000 0.0000000) *
```

```
##   3) upon< 0.019 16  1 Madison (0.0625000 0.9375000) *
```

```
fancyRpartPlot(papers_model$finalModel)
```





Rattle 2021–Nov–09 22:11:37 admin

```
summary(papers_model$finalModel)
```

```
## Call:
## (function (formula, data, weights, subset, na.action = na.rpart,
##     method, model = FALSE, x = FALSE, y = TRUE, parms, control,
##     cost, ...)
## {
##     Call <- match.call()
##     if (is.data.frame(model)) {
##         m <- model
##         model <- FALSE
##     }
##     else {
##         indx <- match(c("formula", "data", "weights", "subset"),
##             names(Call), nomatch = 0)
##         if (indx[1] == 0)
##             stop("a 'formula' argument is required")
##         temp <- Call[c(1, indx)]
##         temp$na.action <- na.action
##         temp[[1]] <- quote(stats::model.frame)
##         m <- eval.parent(temp)
##     }
##     Terms <- attr(m, "terms")
##     if (any(attr(Terms, "order") > 1))
##         stop("Trees cannot handle interaction terms")
## }
```

```

##      Y <- model.response(m)
##      wt <- model.weights(m)
##      if (any(wt < 0))
##          stop("negative weights not allowed")
##      if (!length(wt))
##          wt <- rep(1, nrow(m))
##      offset <- model.offset(m)
##      X <- rpart.matrix(m)
##      nobs <- nrow(X)
##      nvar <- ncol(X)
##      if (missing(method)) {
##          method <- if (is.factor(Y) || is.character(Y))
##              "class"
##          else if (inherits(Y, "Surv"))
##              "exp"
##          else if (is.matrix(Y))
##              "poisson"
##          else "anova"
##      }
##      if (is.list(method)) {
##          mlist <- method
##          method <- "user"
##          init <- if (missing(parms))
##              mlist$init(Y, offset, wt = wt)
##          else mlist$init(Y, offset, parms, wt)
##          keep <- rpartcallback(mlist, nobs, init)
##          method.int <- 4
##          parms <- init$parms
##      }
##      else {
##          method.int <- pmatch(method, c("anova", "poisson", "class",
##              "exp"))
##          if (is.na(method.int))
##              stop("Invalid method")
##          method <- c("anova", "poisson", "class", "exp")[method.int]
##          if (method.int == 4)
##              method.int <- 2
##          init <- if (missing(parms))
##              get(paste("rpart", method, sep = "."), envir = environment())(Y,
##                  offset, , wt)
##          else get(paste("rpart", method, sep = "."), envir = environment())(Y,
##                  offset, parms, wt)
##          ns <- asNamespace("rpart")
##          if (!is.null(init$print))
##              environment(init$print) <- ns
##          if (!is.null(init$summary))
##              environment(init$summary) <- ns
##          if (!is.null(init$text))
##              environment(init$text) <- ns
##      }
##      Y <- init$y
##      xlevels <- .getXlevels(Terms, m)
##      cats <- rep(0, ncol(X))
##      if (!is.null(xlevels))

```

```

##      cats[match(names(xlevels), colnames(X))] <- unlist(lapply(xlevels,
##      length))
##      extraArgs <- list(...)
##      if (length(extraArgs)) {
##          controlargs <- names(formals(rpart.control))
##          indx <- match(names(extraArgs), controlargs, nomatch = 0)
##          if (any(indx == 0))
##              stop(gettextf("Argument %s not matched", names(extraArgs)[indx ==
##              0]), domain = NA)
##      }
##      controls <- rpart.control(...)
##      if (!missing(control))
##          controls[names(control)] <- control
##      xval <- controls$xval
##      if (is.null(xval) || (length(xval) == 1 && xval == 0) ||
##          method == "user") {
##          xgroups <- 0
##          xval <- 0
##      }
##      else if (length(xval) == 1) {
##          xgroups <- sample(rep(1:xval, length = nobs), nobs, replace = FALSE)
##      }
##      else if (length(xval) == nobs) {
##          xgroups <- xval
##          xval <- length(unique(xgroups))
##      }
##      else {
##          if (!is.null(attr(m, "na.action"))) {
##              temp <- as.integer(attr(m, "na.action"))
##              xval <- xval[-temp]
##              if (length(xval) == nobs) {
##                  xgroups <- xval
##                  xval <- length(unique(xgroups))
##              }
##              else stop("Wrong length for 'xval'")
##          }
##          else stop("Wrong length for 'xval'")
##      }
##      if (missing(cost))
##          cost <- rep(1, nvar)
##      else {
##          if (length(cost) != nvar)
##              stop("Cost vector is the wrong length")
##          if (any(cost <= 0))
##              stop("Cost vector must be positive")
##      }
##      tfun <- function(x) if (is.matrix(x))
##          rep(is.ordered(x), ncol(x))
##      else is.ordered(x)
##      labs <- sub("^'(.*)'$", "\\1", attr(Terms, "term.labels"))
##      isord <- unlist(lapply(m[labs], tfun))
##      storage.mode(X) <- "double"
##      storage.mode(wt) <- "double"
##      temp <- as.double(unlist(init$params))

```

```

##   if (!length(temp))
##       temp <- 0
##   rpfit <- .Call(C_rpart, ncat = as.integer(cats * !isord),
##       method = as.integer(method.int), as.double(unlist(controls)),
##       temp, as.integer(xval), as.integer(xgroups), as.double(t(init$y)),
##       X, wt, as.integer(init$numy), as.double(cost))
##   nsplit <- nrow(rpfit$split)
##   ncat <- if (!is.null(rpfit$csplit))
##       nrow(rpfit$csplit)
##   else 0
##   if (nsplit == 0)
##       xval <- 0
##   numcp <- ncol(rpfit$cptable)
##   temp <- if (nrow(rpfit$cptable) == 3)
##       c("CP", "nsplit", "rel error")
##   else c("CP", "nsplit", "rel error", "xerror", "xstd")
##   dimnames(rpfit$cptable) <- list(temp, 1:numcp)
##   tname <- c("<leaf>", colnames(X))
##   splits <- matrix(c(rpfit$split[, 2:3], rpfit$dsplit), ncol = 5,
##       dimnames = list(tname[rpfit$split[, 1] + 1], c("count",
##           "ncat", "improve", "index", "adj")))
##   index <- rpfit$inode[, 2]
##   nadd <- sum(isord[rpfit$split[, 1]])
##   if (nadd > 0) {
##       newc <- matrix(0, nadd, max(cats))
##       cvar <- rpfit$split[, 1]
##       indx <- isord[cvar]
##       cdir <- splits[indx, 2]
##       ccut <- floor(splits[indx, 4])
##       splits[indx, 2] <- cats[cvar[indx]]
##       splits[indx, 4] <- ncat + 1:nadd
##       for (i in 1:nadd) {
##           newc[i, 1:(cats[(cvar[indx])[i]])] <- -as.integer(cdir[i])
##           newc[i, 1:ccut[i]] <- as.integer(cdir[i])
##       }
##       catmat <- if (ncat == 0)
##           newc
##       else {
##           cs <- rpfit$csplit
##           ncs <- ncol(cs)
##           ncc <- ncol(newc)
##           if (ncs < ncc)
##               cs <- cbind(cs, matrix(0, nrow(cs), ncc - ncs))
##           rbind(cs, newc)
##       }
##       ncat <- ncat + nadd
##   }
##   else catmat <- rpfit$csplit
##   if (nsplit == 0) {
##       frame <- data.frame(row.names = 1, var = "<leaf>", n = rpfit$inode[,
##           5], wt = rpfit$dnode[, 3], dev = rpfit$dnode[, 1],
##           yval = rpfit$dnode[, 4], complexity = rpfit$dnode[,
##           2], ncompete = 0, nsurrogate = 0)
##   }

```

```

## else {
##     temp <- ifelse(index == 0, 1, index)
##     svar <- ifelse(index == 0, 0, rpfif$split[temp, 1])
##     frame <- data.frame(row.names = rpfif$inode[, 1], var = tname[svar +
##         1], n = rpfif$inode[, 5], wt = rpfif$dnode[, 3],
##         dev = rpfif$dnode[, 1], yval = rpfif$dnode[, 4],
##         complexity = rpfif$dnode[, 2], ncompete = pmax(0,
##             rpfif$inode[, 3] - 1), nsurrogate = rpfif$inode[,
##             4])
## }
## if (method.int == 3) {
##     numclass <- init$numresp - 2
##     nodeprob <- rpfif$dnode[, numclass + 5]/sum(wt)
##     temp <- pmax(1, init$counts)
##     temp <- rpfif$dnode[, 4 + (1:numclass)] %*% diag(init$parms$prior/temp)
##     yprob <- temp/rowSums(temp)
##     yval2 <- matrix(rpfif$dnode[, 4 + (0:numclass)], ncol = numclass +
##         1)
##     frame$yval2 <- cbind(yval2, yprob, nodeprob)
## }
## else if (init$numresp > 1)
##     frame$yval2 <- rpfif$dnode[, -(1:3), drop = FALSE]
## if (is.null(init$summary))
##     stop("Initialization routine is missing the 'summary' function")
## functions <- if (is.null(init$print))
##     list(summary = init$summary)
## else list(summary = init$summary, print = init$print)
## if (!is.null(init$text))
##     functions <- c(functions, list(text = init$text))
## if (method == "user")
##     functions <- c(functions, mlist)
## where <- rpfif$which
## names(where) <- row.names(m)
## ans <- list(frame = frame, where = where, call = Call, terms = Terms,
##     cptable = t(rpfif$cptable), method = method, parms = init$parms,
##     control = controls, functions = functions, numresp = init$numresp)
## if (nsplit)
##     ans$splits = splits
## if (ncat > 0)
##     ans$csplit <- catmat + 2
## if (nsplit)
##     ans$variable.importance <- importance(ans)
## if (model) {
##     ans$model <- m
##     if (missing(y))
##         y <- FALSE
## }
## if (y)
##     ans$y <- Y
## if (x) {
##     ans$x <- X
##     ans$wt <- wt
## }
## ans$ordered <- isord

```

```

##   if (!is.null(attr(m, "na.action")))
##       ans$na.action <- attr(m, "na.action")
##   if (!is.null(xlevels))
##       attr(ans, "xlevels") <- xlevels
##   if (method == "class")
##       attr(ans, "ylevels") <- init$ylevels
##   class(ans) <- "rpart"
##   ans
## }(formula = .outcome ~ ., data = list(c(0.213, 0.369, 0.305,
## 0.391, 0.327, 0.26, 0.261, 0.449, 0.392, 0.194, 0.361, 0.329,
## 0.329, 0.27, 0.299, 0.325, 0.263, 0.196, 0.344, 0.381, 0.284,
## 0.296, 0.236, 0.382, 0.323, 0.349, 0.315, 0.311, 0.303, 0.203,
## 0.288, 0.323, 0.281, 0.298, 0.378, 0.378, 0.375, 0.379, 0.402,
## 0.358, 0.466, 0.275, 0.27, 0.318, 0.215, 0.32, 0.211, 0.354,
## 0.248, 0.371, 0.328, 0.359, 0.221, 0.29, 0.378, 0.3, 0.25, 0.302,
## 0.24, 0.38, 0.212, 0.136, 0.212, 0.177, 0.243, 0.347), c(0.083,
## 0.07, 0.047, 0.045, 0.096, 0.065, 0.108, 0.022, 0.05, 0.081,
## 0.059, 0.037, 0.051, 0.04, 0.127, 0.027, 0.038, 0.077, 0.059,
## 0.071, 0.041, 0.046, 0.055, 0.023, 0.035, 0.066, 0.02, 0.03,
## 0.053, 0.035, 0.08, 0.086, 0.075, 0.028, 0.017, 0.029, 0.044,
## 0.015, 0.039, 0.023, 0.068, 0.034, 0.028, 0.064, 0.108, 0.042,
## 0.057, 0.068, 0.05, 0.072, 0.058, 0.02, 0.034, 0.048, 0.032,
## 0.074, 0.043, 0.063, 0.042, 0.051, 0.101, 0.054, 0.028, 0.052,
## 0.091, 0.097), c(0, 0.006, 0.007, 0.015, 0, 0, 0, 0, 0.004, 0,
## 0.007, 0.007, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.008, 0, 0, 0.01,
## 0.007, 0.013, 0, 0.02, 0.01, 0, 0, 0.009, 0, 0.006, 0.029, 0.008,
## 0.008, 0.015, 0.005, 0, 0.007, 0.006, 0, 0, 0.005, 0.014, 0.011,
## 0.007, 0, 0.007, 0.016, 0.014, 0.006, 0.014, 0.004, 0, 0.013,
## 0.01, 0.014, 0.006, 0.047, 0.008, 0.007), c(0.083, 0.076, 0.068,
## 0.03, 0.086, 0.087, 0.072, 0.074, 0.075, 0.089, 0.066, 0.052,
## 0.133, 0.05, 0.109, 0.095, 0.053, 0.102, 0.138, 0.071, 0.088,
## 0.086, 0.071, 0.179, 0.07, 0.079, 0.049, 0.081, 0.072, 0.071,
## 0.089, 0.071, 0.094, 0.085, 0.077, 0.073, 0.051, 0.102, 0.077,
## 0.101, 0.143, 0.077, 0.014, 0.085, 0.048, 0.085, 0.096, 0.052,
## 0.054, 0.105, 0.087, 0.069, 0.062, 0.048, 0.072, 0.034, 0.058,
## 0.084, 0.078, 0.051, 0.051, 0.048, 0.05, 0.047, 0.084, 0.056),
##   c(0.343, 0.411, 0.386, 0.27, 0.356, 0.274, 0.467, 0.353,
##   0.329, 0.413, 0.42, 0.329, 0.297, 0.32, 0.308, 0.379, 0.338,
##   0.358, 0.374, 0.372, 0.311, 0.408, 0.33, 0.257, 0.513, 0.217,
##   0.217, 0.274, 0.25, 0.406, 0.258, 0.505, 0.319, 0.36, 0.335,
##   0.364, 0.261, 0.336, 0.247, 0.358, 0.278, 0.338, 0.327, 0.381,
##   0.37, 0.328, 0.374, 0.302, 0.292, 0.388, 0.313, 0.59, 0.407,
##   0.523, 0.401, 0.346, 0.461, 0.369, 0.491, 0.333, 0.44, 0.422,
##   0.391, 0.436, 0.372, 0.313), c(0.056, 0.023, 0.047, 0.045,
##   0.014, 0.079, 0.018, 0.044, 0.029, 0.065, 0.044, 0.03, 0.063,
##   0.07, 0.018, 0.041, 0.06, 0.043, 0.049, 0.062, 0.047, 0.053,
##   0.055, 0.086, 0, 0.079, 0.049, 0.022, 0.046, 0.018, 0.08,
##   0.03, 0.044, 0.066, 0.06, 0.015, 0.07, 0.044, 0.015, 0.039,
##   0.045, 0.043, 0.114, 0.014, 0.036, 0.058, 0.01, 0.042, 0.108,
##   0.067, 0.029, 0.02, 0.021, 0.011, 0.041, 0.023, 0.01, 0.025,
##   0.057, 0.034, 0.076, 0.027, 0.033, 0.026, 0.008, 0.035),
##   c(0.111, 0.053, 0.102, 0.06, 0.086, 0.022, 0.045, 0.059,
##   0.075, 0.138, 0.074, 0.105, 0.038, 0.1, 0.054, 0.061, 0.053,
##   0.154, 0.02, 0.071, 0.129, 0.099, 0.088, 0.047, 0.105, 0.066,

```

```

## 0.089, 0.059, 0.026, 0.062, 0.04, 0.025, 0.044, 0.099, 0.043,
## 0.051, 0.051, 0.044, 0.054, 0.023, 0.045, 0.063, 0.085, 0.099,
## 0.119, 0.058, 0.105, 0.112, 0.115, 0.072, 0.102, 0.133, 0.048,
## 0.095, 0.081, 0.113, 0.062, 0.063, 0.047, 0.064, 0.066, 0.048,
## 0.073, 0.135, 0.046, 0.049), c(0.093, 0.117, 0.108, 0.09,
## 0.072, 0.13, 0.027, 0.133, 0.104, 0.186, 0.088, 0.165, 0.088,
## 0.08, 0.091, 0.176, 0.165, 0.128, 0.03, 0.115, 0.122, 0.145,
## 0.154, 0.117, 0.133, 0.125, 0.167, 0.111, 0.132, 0.132, 0.169,
## 0.081, 0.119, 0.099, 0.138, 0.109, 0.07, 0.117, 0.124, 0.117,
## 0.113, 0.13, 0.057, 0.113, 0.108, 0.177, 0.144, 0.127, 0.119,
## 0.111, 0.138, 0.093, 0.159, 0.148, 0.126, 0.187, 0.144, 0.13,
## 0.115, 0.154, 0.136, 0.15, 0.117, 0.083, 0.137, 0.132), c(0.065,
## 0.065, 0.088, 0.015, 0.115, 0.079, 0.063, 0.029, 0.05, 0.024,
## 0.052, 0.082, 0.063, 0.04, 0.063, 0.068, 0.038, 0.043, 0.02,
## 0.018, 0.041, 0.026, 0.016, 0.062, 0.042, 0.053, 0.118, 0.044,
## 0.007, 0.035, 0.06, 0.03, 0.063, 0.057, 0.069, 0.087, 0.064,
## 0.029, 0.015, 0.031, 0.045, 0.014, 0.071, 0.078, 0.018, 0.027,
## 0.019, 0.047, 0.036, 0.033, 0.073, 0.039, 0.048, 0.011, 0.027,
## 0.04, 0.034, 0.021, 0.016, 0.026, 0.02, 0.027, 0.033, 0.036,
## 0.03, 0.035), c(0.315, 0.258, 0.271, 0.376, 0.211, 0.397,
## 0.216, 0.295, 0.221, 0.356, 0.383, 0.389, 0.335, 0.34, 0.371,
## 0.339, 0.36, 0.307, 0.295, 0.284, 0.284, 0.296, 0.434, 0.343,
## 0.126, 0.441, 0.365, 0.481, 0.257, 0.141, 0.358, 0.258, 0.288,
## 0.18, 0.327, 0.298, 0.305, 0.35, 0.216, 0.452, 0.286, 0.319,
## 0.427, 0.247, 0.287, 0.328, 0.326, 0.273, 0.338, 0.188, 0.189,
## 0.3, 0.31, 0.248, 0.23, 0.295, 0.259, 0.311, 0.25, 0.333,
## 0.273, 0.354, 0.402, 0.182, 0.235, 0.361), c(0.028, 0.018,
## 0.054, 0.03, 0.067, 0.051, 0.027, 0.081, 0.067, 0.049, 0.029,
## 0.045, 0.051, 0.06, 0.036, 0.041, 0.075, 0.043, 0.039, 0.098,
## 0.081, 0.033, 0.071, 0.117, 0.084, 0.013, 0.089, 0.089, 0.066,
## 0.15, 0.05, 0.04, 0.05, 0.071, 0.043, 0.029, 0.089, 0.029,
## 0.077, 0.062, 0.068, 0.043, 0.071, 0.113, 0.084, 0.085, 0.01,
## 0.094, 0.065, 0.094, 0.109, 0.044, 0.062, 0.127, 0.068, 0.045,
## 0.096, 0.084, 0.099, 0.06, 0.096, 0.041, 0.028, 0.073, 0.091,
## 0.097), c(0, 0.023, 0.041, 0.03, 0.034, 0.036, 0.009, 0.015,
## 0.05, 0.024, 0.037, 0.022, 0.038, 0.04, 0.027, 0.027, 0.045,
## 0.026, 0.089, 0.08, 0.034, 0.033, 0.033, 0.039, 0.014, 0.02,
## 0, 0.015, 0.026, 0.035, 0.05, 0.025, 0.006, 0.024, 0.026,
## 0.022, 0.032, 0, 0.031, 0.016, 0.023, 0.058, 0.014, 0.056,
## 0.012, 0.042, 0.019, 0.042, 0.047, 0.022, 0.029, 0.044, 0.021,
## 0.048, 0.077, 0.057, 0.019, 0.025, 0.026, 0.03, 0.035, 0.02,
## 0.045, 0.026, 0.023, 0.035), c(0.13, 0.106, 0.095, 0.075,
## 0.154, 0.094, 0.081, 0.162, 0.125, 0.089, 0.103, 0.165, 0.133,
## 0.14, 0.063, 0.047, 0.105, 0.077, 0.098, 0.124, 0.027, 0.066,
## 0.11, 0.132, 0.077, 0.138, 0.049, 0.118, 0.086, 0.088, 0.119,
## 0.111, 0.175, 0.071, 0.129, 0.116, 0.172, 0.029, 0.124, 0.156,
## 0.12, 0.116, 0.057, 0.078, 0.072, 0.123, 0.038, 0.213, 0.108,
## 0.055, 0.095, 0.177, 0.124, 0.158, 0.162, 0.187, 0.264, 0.139,
## 0.162, 0.205, 0.142, 0.075, 0.112, 0.218, 0.213, 0.153),
## c(0.028, 0.029, 0.014, 0.06, 0.067, 0.007, 0.018, 0.066,
## 0.025, 0.089, 0.022, 0.022, 0.019, 0.11, 0.109, 0.075, 0.083,
## 0.017, 0.01, 0.018, 0.047, 0.04, 0.088, 0.047, 0.014, 0.026,
## 0.049, 0.037, 0.02, 0.026, 0.01, 0.091, 0.019, 0.024, 0.017,
## 0.007, 0.019, 0, 0, 0.008, 0.06, 0.087, 0.1, 0.007, 0.03,

```

```

## 0.023, 0.048, 0.016, 0.04, 0.055, 0.007, 0.02, 0.055, 0.011,
## 0.041, 0.006, 0.014, 0.038, 0.016, 0.038, 0.035, 0, 0.039,
## 0.047, 0.03, 0.035), c(0.009, 0.012, 0, 0.015, 0.019, 0.014,
## 0, 0.015, 0.008, 0, 0.015, 0.007, 0, 0, 0.009, 0, 0, 0.009,
## 0, 0.018, 0, 0.007, 0.005, 0.008, 0, 0, 0.01, 0, 0.007, 0,
## 0.01, 0.015, 0, 0.009, 0.009, 0, 0.013, 0, 0, 0.008, 0.008,
## 0.014, 0.014, 0, 0.006, 0.004, 0.01, 0.01, 0.014, 0.006,
## 0, 0, 0, 0.005, 0.005, 0, 0, 0.008, 0.005, 0.009, 0.01, 0,
## 0.006, 0.005, 0, 0.007), c(0, 0, 0, 0, 0.005, 0.007, 0, 0,
## 0, 0, 0, 0, 0, 0.01, 0, 0, 0, 0, 0, 0, 0.007, 0.013, 0, 0,
## 0, 0, 0, 0.007, 0, 0, 0, 0, 0, 0, 0.017, 0, 0, 0, 0, 0.008,
## 0, 0.005, 0, 0, 0.012, 0, 0.01, 0, 0, 0, 0, 0, 0, 0, 0, 0,
## 0, 0.004, 0, 0, 0, 0, 0, 0.01, 0, 0), c(0.019, 0.018, 0.014,
## 0, 0.014, 0.029, 0, 0.015, 0.021, 0, 0.037, 0.022, 0.032,
## 0, 0.018, 0.034, 0.008, 0.017, 0.01, 0, 0.014, 0.013, 0.027,
## 0, 0, 0, 0, 0, 0.02, 0.035, 0.01, 0.015, 0.019, 0.009, 0.009,
## 0.036, 0.013, 0, 0, 0.016, 0, 0.01, 0, 0.007, 0.018, 0.012,
## 0.01, 0.008, 0.011, 0, 0.007, 0, 0.014, 0.011, 0.005, 0,
## 0.024, 0.017, 0.005, 0.013, 0.005, 0.007, 0.011, 0.021, 0,
## 0.014), c(0.028, 0.012, 0.027, 0.045, 0.038, 0.007, 0.009,
## 0.022, 0.029, 0.024, 0.007, 0.007, 0.025, 0, 0, 0, 0.053,
## 0.034, 0.02, 0.035, 0.014, 0.026, 0.011, 0.047, 0.007, 0.013,
## 0.01, 0.022, 0.02, 0, 0.02, 0.02, 0.006, 0.009, 0.026, 0.022,
## 0.025, 0.029, 0.008, 0.031, 0, 0.034, 0.028, 0.021, 0.012,
## 0.035, 0.038, 0.029, 0.022, 0.039, 0.015, 0.01, 0.007, 0.016,
## 0.009, 0.034, 0.038, 0.055, 0.037, 0.038, 0.066, 0.034, 0.022,
## 0.021, 0.023, 0.056), c(0.093, 0.106, 0.054, 0.03, 0.086,
## 0.065, 0.099, 0.037, 0.088, 0.13, 0.074, 0.075, 0.133, 0.07,
## 0.127, 0.115, 0.173, 0.051, 0.039, 0.08, 0.176, 0.072, 0.066,
## 0.101, 0.084, 0.079, 0.167, 0.118, 0.112, 0.071, 0.149, 0.091,
## 0.081, 0.114, 0.086, 0.102, 0.083, 0.131, 0.054, 0.062, 0.143,
## 0.068, 0.213, 0.064, 0.072, 0.081, 0.038, 0.088, 0.09, 0.061,
## 0.073, 0.088, 0.138, 0.063, 0.095, 0.079, 0.111, 0.143, 0.11,
## 0.119, 0.101, 0.082, 0.056, 0.036, 0.114, 0.083), c(0.102,
## 0.111, 0.129, 0.075, 0.101, 0.079, 0.045, 0.118, 0.096, 0.049,
## 0.074, 0.09, 0.063, 0.05, 0.054, 0.088, 0.053, 0.085, 0.089,
## 0.08, 0.081, 0.053, 0.121, 0.093, 0.098, 0.079, 0.079, 0.089,
## 0.072, 0.044, 0.109, 0.061, 0.069, 0.081, 0.12, 0.102, 0.076,
## 0.044, 0.077, 0.078, 0.098, 0.13, 0.1, 0.064, 0.036, 0.073,
## 0.125, 0.049, 0.065, 0.05, 0.044, 0.049, 0.076, 0.1, 0.086,
## 0.125, 0.062, 0.092, 0.052, 0.026, 0.045, 0.054, 0.05, 0.057,
## 0.091, 0.063), c(0.009, 0.006, 0.02, 0, 0.01, 0.036, 0.018,
## 0.007, 0.013, 0.008, 0.037, 0.03, 0.044, 0, 0.027, 0.014,
## 0, 0, 0.02, 0.027, 0.014, 0.013, 0.016, 0.016, 0.028, 0,
## 0.03, 0.022, 0.013, 0.026, 0, 0.005, 0.044, 0.024, 0.034,
## 0.051, 0.013, 0.044, 0.008, 0.016, 0.023, 0.014, 0, 0.021,
## 0.006, 0.015, 0, 0.013, 0.007, 0.011, 0.036, 0.005, 0.014,
## 0.016, 0.027, 0, 0.067, 0.017, 0.021, 0.013, 0.02, 0.014,
## 0.017, 0.016, 0.076, 0.014), c(0.056, 0.047, 0.088, 0.03,
## 0.062, 0.051, 0.036, 0.066, 0.083, 0.041, 0.066, 0.037, 0.038,
## 0.02, 0.045, 0.034, 0.068, 0.009, 0.02, 0.027, 0.041, 0.02,
## 0.049, 0.047, 0.049, 0.02, 0, 0.007, 0.072, 0.106, 0.06,
## 0.061, 0.025, 0.047, 0.043, 0.036, 0.089, 0.029, 0.054, 0.039,
## 0.06, 0.029, 0.114, 0.056, 0.036, 0.046, 0, 0.057, 0.068,

```



```

## 0.083, 0.087, 0.02, 0.055, 0.053, 0.054, 0.051, 0.048, 0.076,
## 0.068, 0.047, 0.035, 0.027, 0.022, 0.088, 0.046, 0.076),
## c(0.093, 0.088, 0.088, 0.06, 0.144, 0.087, 0.036, 0.066,
## 0.058, 0.081, 0.088, 0.082, 0.101, 0.05, 0.1, 0.047, 0.083,
## 0.094, 0.049, 0.115, 0.108, 0.059, 0.121, 0.171, 0.169, 0.04,
## 0.118, 0.126, 0.086, 0.185, 0.139, 0.086, 0.144, 0.137, 0.077,
## 0.051, 0.089, 0.073, 0.124, 0.125, 0.098, 0.092, 0.085, 0.127,
## 0.114, 0.089, 0.067, 0.14, 0.094, 0.127, 0.175, 0.079, 0.11,
## 0.137, 0.095, 0.017, 0.125, 0.109, 0.104, 0.09, 0.167, 0.116,
## 0.084, 0.073, 0.076, 0.125), c(0, 0.076, 0.007, 0.15, 0,
## 0, 0, 0.007, 0.004, 0, 0.007, 0.022, 0, 0, 0.027, 0, 0, 0,
## 0, 0, 0, 0.007, 0, 0, 0.014, 0, 0, 0, 0, 0, 0, 0, 0.019,
## 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.007, 0, 0, 0, 0.013, 0, 0,
## 0, 0, 0.021, 0.005, 0.032, 0, 0, 0.055, 0, 0.004, 0, 0, 0,
## 0, 0, 0), c(0, 0.006, 0.014, 0, 0.005, 0, 0.063, 0.015, 0.029,
## 0, 0.037, 0, 0.038, 0.03, 0, 0.014, 0.015, 0, 0, 0, 0, 0.066,
## 0, 0, 0.021, 0, 0.01, 0.081, 0, 0.044, 0.06, 0.081, 0, 0.062,
## 0.163, 0.247, 0.159, 0.015, 0.062, 0.125, 0.045, 0.01, 0.014,
## 0, 0.012, 0, 0, 0.003, 0.014, 0.039, 0.029, 0.039, 0, 0.026,
## 0.081, 0.006, 0.014, 0.013, 0.005, 0, 0.01, 0.007, 0, 0.052,
## 0.023, 0.007), c(0.028, 0.029, 0.027, 0.045, 0.014, 0.058,
## 0.018, 0.044, 0.025, 0.016, 0.044, 0, 0.082, 0.03, 0.036,
## 0.041, 0.023, 0.017, 0.049, 0.027, 0.034, 0.033, 0.022, 0.023,
## 0.021, 0.02, 0.02, 0.007, 0.079, 0.026, 0.04, 0.035, 0.056,
## 0.028, 0.06, 0.029, 0.013, 0.029, 0.008, 0.008, 0.015, 0.029,
## 0.028, 0.028, 0.018, 0.019, 0.01, 0.049, 0.022, 0.011, 0.007,
## 0.02, 0.014, 0.005, 0.023, 0.011, 0.034, 0.038, 0.016, 0.021,
## 0.03, 0.034, 0.017, 0.016, 0.008, 0.035), c(0.232, 0.387,
## 0.311, 0.24, 0.317, 0.274, 0.252, 0.331, 0.321, 0.211, 0.406,
## 0.285, 0.398, 0.31, 0.353, 0.257, 0.33, 0.401, 0.384, 0.284,
## 0.339, 0.296, 0.341, 0.475, 0.372, 0.494, 0.463, 0.355, 0.454,
## 0.291, 0.398, 0.323, 0.244, 0.374, 0.318, 0.255, 0.375, 0.336,
## 0.34, 0.35, 0.413, 0.309, 0.47, 0.282, 0.269, 0.494, 0.364,
## 0.499, 0.306, 0.394, 0.262, 0.29, 0.269, 0.311, 0.27, 0.351,
## 0.298, 0.248, 0.303, 0.213, 0.298, 0.252, 0.251, 0.27, 0.304,
## 0.368), c(0.019, 0.023, 0.047, 0.105, 0.019, 0.022, 0.027,
## 0, 0.017, 0.016, 0.007, 0.007, 0.051, 0.03, 0.009, 0.027,
## 0.045, 0.043, 0.01, 0.035, 0.014, 0.033, 0.022, 0.039, 0.049,
## 0.033, 0.03, 0.022, 0.013, 0.009, 0, 0.04, 0.013, 0.014,
## 0, 0, 0.025, 0.015, 0.015, 0, 0, 0.01, 0.014, 0.035, 0.006,
## 0.008, 0.01, 0.005, 0.007, 0.011, 0.036, 0.044, 0.014, 0.026,
## 0.032, 0.017, 0.034, 0.029, 0.042, 0.03, 0.056, 0.02, 0.017,
## 0.005, 0.023, 0.014), c(0.12, 0.053, 0.149, 0.135, 0.207,
## 0.101, 0.099, 0.199, 0.192, 0.154, 0.074, 0.127, 0.139, 0.14,
## 0.118, 0.203, 0.113, 0.196, 0.157, 0.231, 0.108, 0.263, 0.187,
## 0.132, 0.028, 0.178, 0.099, 0.111, 0.191, 0.159, 0.189, 0.258,
## 0.081, 0.232, 0.12, 0.175, 0.184, 0.16, 0.131, 0.156, 0.158,
## 0.237, 0.157, 0.148, 0.137, 0.235, 0.144, 0.299, 0.238, 0.211,
## 0.117, 0.201, 0.166, 0.158, 0.216, 0.323, 0.139, 0.143, 0.204,
## 0.141, 0.142, 0.15, 0.039, 0.218, 0.213, 0.118), c(0.139,
## 0.088, 0.142, 0.09, 0.096, 0.188, 0.09, 0.103, 0.146, 0.154,
## 0.133, 0.15, 0.209, 0.15, 0.145, 0.23, 0.173, 0.179, 0.197,
## 0.284, 0.129, 0.151, 0.176, 0.195, 0.077, 0.25, 0.217, 0.2,
## 0.197, 0.15, 0.129, 0.131, 0.075, 0.152, 0.206, 0.145, 0.178,

```

```

## 0.175, 0.162, 0.14, 0.18, 0.193, 0.142, 0.12, 0.137, 0.208,
## 0.086, 0.213, 0.176, 0.188, 0.124, 0.197, 0.179, 0.158, 0.198,
## 0.215, 0.144, 0.164, 0.12, 0.196, 0.081, 0.116, 0.151, 0.088,
## 0.129, 0.111), c(0.074, 0.035, 0.047, 0.03, 0.038, 0.116,
## 0.054, 0.081, 0.067, 0.089, 0.029, 0.067, 0.025, 0.15, 0.072,
## 0.041, 0.15, 0.068, 0.03, 0.115, 0.074, 0.02, 0.093, 0.093,
## 0, 0.066, 0.03, 0.022, 0.026, 0.026, 0.03, 0.035, 0.013,
## 0.038, 0.026, 0.051, 0.057, 0.044, 0.046, 0.047, 0.038, 0.058,
## 0.028, 0.071, 0.048, 0.035, 0.038, 0.055, 0.025, 0.033, 0.036,
## 0.044, 0.048, 0.037, 0.027, 0.113, 0.019, 0.063, 0.047, 0.064,
## 0.03, 0.041, 0.033, 0.052, 0.061, 0.042), c(0.102, 0.053,
## 0.02, 0.12, 0.034, 0.036, 0.036, 0.081, 0.083, 0.049, 0.029,
## 0.022, 0.063, 0.04, 0.063, 0.047, 0.068, 0.085, 0.02, 0.089,
## 0.041, 0.053, 0.082, 0.132, 0.021, 0.066, 0.108, 0.052, 0.053,
## 0.026, 0.05, 0.051, 0.019, 0.076, 0.086, 0.044, 0.076, 0.102,
## 0.031, 0.055, 0.015, 0.077, 0.085, 0.056, 0.084, 0.096, 0.134,
## 0.042, 0.094, 0.089, 0.036, 0.079, 0.11, 0.095, 0.041, 0.034,
## 0.024, 0.097, 0.089, 0.094, 0.03, 0.061, 0.073, 0.042, 0.046,
## 0.069), c(0.065, 0.053, 0.034, 0.09, 0.029, 0.043, 0.081,
## 0.022, 0.025, 0.016, 0.007, 0.015, 0.019, 0.08, 0.081, 0.014,
## 0.023, 0.009, 0, 0.009, 0.027, 0.059, 0.055, 0.07, 0.056,
## 0.026, 0.039, 0.052, 0.079, 0.026, 0.02, 0.035, 0.038, 0.081,
## 0.052, 0.073, 0.064, 0.015, 0.031, 0.039, 0.053, 0.019, 0.071,
## 0.056, 0.012, 0.015, 0.01, 0.06, 0.018, 0.028, 0.044, 0.059,
## 0.055, 0.037, 0.045, 0.034, 0.019, 0.05, 0.005, 0.034, 0.051,
## 0.095, 0.061, 0.047, 0.061, 0.09), c(0.028, 0.012, 0.108,
## 0.045, 0.043, 0.065, 0.009, 0.029, 0.038, 0.073, 0.044, 0.037,
## 0.025, 0.02, 0.063, 0.007, 0.09, 0.068, 0.01, 0.062, 0.027,
## 0.007, 0.055, 0.016, 0.007, 0.04, 0.02, 0.015, 0.033, 0.026,
## 0.01, 0.025, 0.019, 0.043, 0.026, 0.044, 0.006, 0, 0.023,
## 0.039, 0.03, 0.068, 0.028, 0.021, 0.006, 0.012, 0.019, 0.039,
## 0.043, 0.089, 0.007, 0.049, 0.014, 0.111, 0.018, 0.028, 0.014,
## 0.063, 0.021, 0.034, 0.03, 0.041, 0.05, 0.005, 0.008, 0.021
## ), c(0.056, 0, 0.007, 0, 0.014, 0, 0.009, 0, 0.004, 0, 0,
## 0, 0, 0, 0, 0.007, 0, 0, 0, 0, 0.007, 0, 0.011, 0, 0, 0,
## 0.01, 0, 0.007, 0, 0, 0, 0, 0.005, 0, 0, 0, 0, 0.008, 0,
## 0.008, 0, 0, 0, 0, 0.008, 0, 0.013, 0.004, 0.039, 0, 0, 0.014,
## 0, 0, 0, 0, 0, 0, 0, 0, 0.007, 0, 0, 0, 0.007), c(0.028,
## 0.029, 0.007, 0.045, 0.029, 0.029, 0, 0.081, 0.021, 0.016,
## 0.015, 0.015, 0.025, 0.03, 0.054, 0.034, 0.038, 0.026, 0.059,
## 0, 0.047, 0.033, 0.022, 0.062, 0, 0.026, 0.039, 0.022, 0.033,
## 0.026, 0.02, 0.051, 0.019, 0.043, 0.009, 0.044, 0.032, 0,
## 0.015, 0.031, 0.015, 0.082, 0.043, 0.021, 0.03, 0.031, 0.029,
## 0.055, 0.083, 0.067, 0.015, 0.02, 0.062, 0.026, 0.054, 0.04,
## 0.038, 0.025, 0.042, 0.06, 0.066, 0.041, 0.022, 0.073, 0.038,
## 0.035), c(0.13, 0.117, 0.041, 0.075, 0.077, 0.137, 0.036,
## 0.059, 0.104, 0.081, 0.088, 0.06, 0.133, 0.02, 0.072, 0.061,
## 0.053, 0.102, 0.148, 0.106, 0.095, 0.125, 0.077, 0.101, 0.112,
## 0.086, 0.079, 0.133, 0.138, 0.097, 0.109, 0.056, 0.088, 0.071,
## 0.06, 0.095, 0.076, 0.058, 0.093, 0.109, 0.045, 0.087, 0.114,
## 0.085, 0.09, 0.096, 0.125, 0.091, 0.108, 0.111, 0.095, 0.069,
## 0.09, 0.095, 0.144, 0.125, 0.091, 0.071, 0.115, 0.111, 0.076,
## 0.068, 0.112, 0.062, 0.099, 0.097), c(0, 0.006, 0.007, 0,
## 0.01, 0.007, 0, 0.022, 0.004, 0.008, 0.022, 0, 0, 0, 0, 0,

```

```

## 0, 0, 0.02, 0, 0.007, 0.007, 0.005, 0, 0.007, 0.007, 0.01,
## 0, 0, 0.009, 0, 0.005, 0, 0.009, 0, 0, 0.006, 0, 0.015, 0,
## 0.008, 0.014, 0, 0.007, 0.006, 0.023, 0.01, 0.016, 0.025,
## 0.017, 0.015, 0, 0.014, 0, 0.009, 0.011, 0.01, 0.013, 0,
## 0.004, 0.01, 0, 0.006, 0, 0, 0), c(0.908, 1.061, 0.961, 0.887,
## 0.947, 1.04, 0.998, 1.119, 1.008, 0.81, 0.877, 0.936, 0.929,
## 0.94, 0.887, 1.016, 0.833, 0.887, 0.895, 0.807, 0.934, 0.863,
## 0.978, 0.888, 1.075, 0.962, 1.005, 0.969, 1.112, 0.927, 0.885,
## 1.162, 0.982, 0.867, 0.851, 0.88, 0.915, 1.211, 1.152, 0.903,
## 1.089, 0.904, 0.783, 0.96, 0.986, 0.961, 0.9, 0.853, 1.03,
## 0.953, 1.13, 0.772, 0.862, 0.85, 0.856, 0.89, 0.769, 0.856,
## 0.96, 0.951, 0.864, 0.98, 0.91, 0.961, 0.744, 0.806), c(0.074,
## 0.029, 0.081, 0.045, 0.048, 0.029, 0.018, 0.044, 0.038, 0.016,
## 0.066, 0.075, 0.044, 0.04, 0.009, 0.02, 0.06, 0.043, 0.118,
## 0.044, 0.074, 0.053, 0.033, 0.047, 0.014, 0.033, 0.059, 0.037,
## 0.046, 0.026, 0.03, 0.03, 0.088, 0.066, 0.06, 0.044, 0.032,
## 0.015, 0.039, 0.031, 0.023, 0.048, 0.071, 0.078, 0.054, 0.062,
## 0, 0.047, 0.072, 0.1, 0.066, 0.084, 0.11, 0.095, 0.063, 0.125,
## 0.062, 0.097, 0.115, 0.137, 0.147, 0.054, 0.156, 0.099, 0.114,
## 0.118), c(0.028, 0.029, 0.027, 0.135, 0.005, 0.014, 0.018,
## 0.007, 0.038, 0.024, 0.059, 0.015, 0.032, 0.01, 0.036, 0.027,
## 0.038, 0.017, 0.039, 0.018, 0.054, 0.026, 0.033, 0.031, 0.021,
## 0.04, 0.03, 0.044, 0.033, 0.026, 0.05, 0.056, 0.025, 0.038,
## 0.06, 0.022, 0.044, 0.044, 0.031, 0.062, 0.03, 0.034, 0.014,
## 0.042, 0.03, 0.039, 0.067, 0.047, 0.014, 0.044, 0.073, 0.039,
## 0.028, 0.032, 0.054, 0.074, 0.038, 0.034, 0.037, 0.055, 0.025,
## 0.027, 0.045, 0.073, 0.053, 0.028), c(0, 0.029, 0, 0, 0.038,
## 0.022, 0, 0.015, 0.004, 0, 0.022, 0.015, 0.032, 0.04, 0.018,
## 0.027, 0.008, 0.026, 0.039, 0.044, 0.02, 0.013, 0.011, 0.016,
## 0.007, 0.007, 0.039, 0.007, 0.02, 0.044, 0.02, 0.035, 0.006,
## 0.019, 0.017, 0.007, 0.019, 0.015, 0.031, 0.023, 0.03, 0.034,
## 0.043, 0.014, 0.006, 0.023, 0.048, 0.023, 0.032, 0.006, 0.022,
## 0.01, 0.028, 0.053, 0.009, 0.034, 0.019, 0.025, 0.037, 0.026,
## 0.02, 0.007, 0.033, 0.01, 0.008, 0.035), c(0.056, 0.07, 0.047,
## 0.045, 0.183, 0.123, 0.117, 0.147, 0.088, 0.049, 0.088, 0.105,
## 0.076, 0.14, 0.072, 0.108, 0.15, 0.111, 0.089, 0.106, 0.074,
## 0.079, 0.154, 0.055, 0.133, 0.105, 0.079, 0.133, 0.079, 0.044,
## 0.06, 0.071, 0.063, 0.104, 0.138, 0.116, 0.153, 0.175, 0.07,
## 0.179, 0.068, 0.092, 0.057, 0.12, 0.131, 0.081, 0.125, 0.068,
## 0.076, 0.067, 0.08, 0.108, 0.076, 0.069, 0.077, 0.079, 0.043,
## 0.088, 0.078, 0.09, 0.182, 0.054, 0.078, 0.088, 0.084, 0.069
## ), c(0.028, 0.199, 0.034, 0, 0.091, 0.007, 0.018, 0.007,
## 0.05, 0.057, 0.088, 0.075, 0.032, 0, 0.009, 0.041, 0.023,
## 0.017, 0, 0.009, 0.041, 0.02, 0.011, 0, 0.028, 0, 0, 0, 0.013,
## 0.009, 0.01, 0.005, 0.031, 0.005, 0, 0, 0.013, 0, 0.015,
## 0.008, 0, 0.01, 0.028, 0.035, 0.006, 0.015, 0.01, 0.016,
## 0.014, 0.028, 0, 0.039, 0.083, 0.032, 0.009, 0.011, 0.005,
## 0.025, 0.01, 0.004, 0, 0.027, 0, 0, 0.023, 0), c(0.065, 0.018,
## 0.007, 0.045, 0.005, 0.007, 0.009, 0.037, 0.008, 0.008, 0.007,
## 0.037, 0.006, 0, 0, 0.02, 0, 0, 0.039, 0.018, 0.02, 0.026,
## 0.016, 0.016, 0.007, 0.013, 0.02, 0, 0.007, 0.044, 0.04,
## 0.03, 0.006, 0.009, 0.009, 0.007, 0.019, 0.015, 0.046, 0.055,
## 0.023, 0.014, 0.057, 0, 0.03, 0.027, 0.077, 0.013, 0.079,
## 0.017, 0.029, 0.005, 0, 0, 0.014, 0, 0.029, 0.017, 0.047,

```

```

## 0.03, 0.076, 0, 0, 0.036, 0.023, 0.021), c(0.009, 0.018,
## 0.027, 0, 0.029, 0.043, 0.018, 0.037, 0.033, 0.016, 0.044,
## 0.037, 0.038, 0.02, 0.072, 0.054, 0.03, 0.051, 0.03, 0.035,
## 0.041, 0.066, 0.027, 0.008, 0.014, 0.033, 0.069, 0.022, 0.033,
## 0.009, 0.05, 0.03, 0.025, 0.028, 0.052, 0, 0.044, 0.029,
## 0, 0.008, 0.015, 0.024, 0.028, 0.035, 0.018, 0.031, 0.048,
## 0.023, 0.004, 0.022, 0.022, 0, 0.014, 0.053, 0.045, 0.006,
## 0.053, 0, 0, 0.038, 0.02, 0.014, 0.05, 0.01, 0.046, 0.007
## ), c(0.019, 0.029, 0.041, 0.045, 0.01, 0.022, 0.036, 0.052,
## 0.038, 0.008, 0.029, 0.007, 0.063, 0.02, 0.009, 0.054, 0.015,
## 0.06, 0, 0.027, 0.02, 0.04, 0.005, 0.023, 0.021, 0.04, 0.02,
## 0.037, 0.039, 0.062, 0.06, 0.01, 0.025, 0.052, 0.043, 0.007,
## 0.032, 0.044, 0.046, 0.016, 0.03, 0.029, 0.014, 0.035, 0.024,
## 0.019, 0.01, 0.018, 0.022, 0.028, 0.022, 0.015, 0.041, 0.042,
## 0.032, 0.028, 0.062, 0.038, 0.042, 0.026, 0.03, 0.007, 0.006,
## 0.021, 0.038, 0.007), c(0, 0.006, 0.014, 0, 0.01, 0.022,
## 0.009, 0.022, 0.017, 0.008, 0.029, 0.037, 0.025, 0.01, 0,
## 0.007, 0.008, 0.026, 0, 0.009, 0.027, 0.007, 0, 0.031, 0.028,
## 0.02, 0.02, 0.03, 0.013, 0.009, 0.01, 0.005, 0.038, 0.005,
## 0.017, 0.022, 0.013, 0, 0.015, 0.031, 0.045, 0.01, 0.028,
## 0.014, 0.048, 0.019, 0.01, 0.021, 0.004, 0.011, 0.007, 0.015,
## 0.048, 0.021, 0.045, 0.011, 0.034, 0.004, 0.005, 0.009, 0.025,
## 0.007, 0.006, 0.016, 0.046, 0.028), c(0.009, 0.029, 0.02,
## 0.015, 0.005, 0.022, 0, 0.022, 0.038, 0.057, 0.029, 0.03,
## 0.057, 0.03, 0.018, 0.027, 0.03, 0.034, 0.049, 0.035, 0.02,
## 0.02, 0.038, 0.023, 0.014, 0.026, 0, 0.03, 0.039, 0.018,
## 0.03, 0.02, 0.019, 0.024, 0.034, 0.051, 0.013, 0.044, 0.015,
## 0.039, 0.03, 0.034, 0.071, 0.035, 0.036, 0.05, 0.038, 0.023,
## 0.029, 0.028, 0.029, 0.039, 0.007, 0.021, 0.05, 0.023, 0.053,
## 0.025, 0.042, 0.051, 0.035, 0.014, 0.033, 0.01, 0, 0.028),
## c(0.102, 0.035, 0.027, 0.15, 0.048, 0.029, 0.045, 0.029,
## 0.046, 0, 0.029, 0.03, 0.032, 0.07, 0.027, 0.014, 0.023,
## 0.051, 0.01, 0.009, 0.02, 0.053, 0.033, 0.055, 0.042, 0.046,
## 0.118, 0.037, 0.046, 0.044, 0.06, 0.061, 0.031, 0.066, 0.043,
## 0.051, 0.089, 0.029, 0.062, 0.055, 0.045, 0.048, 0.1, 0.021,
## 0.024, 0.023, 0.029, 0.042, 0.058, 0.028, 0.007, 0.064, 0.034,
## 0.026, 0.05, 0.045, 0.024, 0.055, 0.021, 0.043, 0.04, 0.061,
## 0.061, 0.026, 0.03, 0.035), c(0.241, 0.141, 0.176, 0.27,
## 0.178, 0.116, 0.189, 0.177, 0.221, 0.227, 0.243, 0.225, 0.265,
## 0.38, 0.172, 0.196, 0.233, 0.256, 0.315, 0.284, 0.284, 0.244,
## 0.225, 0.226, 0.155, 0.303, 0.315, 0.17, 0.237, 0.141, 0.189,
## 0.227, 0.169, 0.152, 0.189, 0.131, 0.248, 0.19, 0.271, 0.265,
## 0.24, 0.324, 0.256, 0.127, 0.149, 0.185, 0.182, 0.309, 0.302,
## 0.188, 0.182, 0.152, 0.221, 0.243, 0.252, 0.153, 0.298, 0.172,
## 0.167, 0.209, 0.142, 0.204, 0.212, 0.171, 0.205, 0.222),
## c(1.094, 1.044, 1.097, 1.052, 1.163, 1.322, 1.366, 1.23,
## 1.125, 1.45, 1.04, 1.205, 1.195, 1.47, 1.43, 1.436, 1.178,
## 1.356, 1.377, 1.339, 1.232, 1.199, 1.275, 1.309, 1.11, 1.416,
## 1.438, 1.539, 1.56, 1.448, 1.352, 1.415, 1.188, 1.303, 1.487,
## 1.258, 1.246, 1.43, 1.515, 1.199, 1.27, 1.435, 1.11, 0.953,
## 1.368, 1.404, 1.465, 1.165, 1.303, 1.275, 1.151, 1.204, 1.352,
## 1.135, 1.18, 1.552, 1.259, 1.293, 1.247, 1.395, 1.501, 1.803,
## 1.613, 1.522, 1.184, 1.389), c(0.13, 0.059, 0.095, 0.015,
## 0.034, 0.087, 0.09, 0.059, 0.092, 0.065, 0.096, 0.082, 0.063,

```

```

## 0.06, 0.118, 0.095, 0.06, 0.085, 0.039, 0.044, 0.068, 0.125,
## 0.077, 0.078, 0.091, 0.092, 0.049, 0.037, 0.092, 0.026, 0.099,
## 0.03, 0.1, 0.104, 0.12, 0.087, 0.051, 0.029, 0.015, 0.055,
## 0.083, 0.155, 0.171, 0.106, 0.036, 0.042, 0.057, 0.039, 0.086,
## 0.039, 0.102, 0.103, 0.131, 0.121, 0.113, 0.091, 0.168, 0.055,
## 0.063, 0.043, 0.045, 0.048, 0.162, 0.005, 0.099, 0.09), c(0,
## 0.012, 0, 0, 0, 0, 0, 0, 0.004, 0, 0, 0.015, 0.019, 0, 0.018,
## 0, 0.008, 0, 0.02, 0.009, 0.007, 0, 0.005, 0, 0, 0.007, 0.01,
## 0, 0.007, 0.009, 0, 0, 0, 0.005, 0.009, 0.007, 0, 0, 0.008,
## 0.008, 0, 0.01, 0.014, 0, 0.006, 0.004, 0.019, 0.003, 0.004,
## 0.011, 0.007, 0.01, 0, 0.005, 0.005, 0.017, 0.019, 0.008,
## 0.005, 0, 0, 0.014, 0, 0.01, 0, 0), c(0.019, 0.023, 0.041,
## 0.105, 0.072, 0.022, 0.018, 0.029, 0.054, 0.032, 0.037, 0.007,
## 0.051, 0.08, 0.063, 0.068, 0.038, 0.051, 0.059, 0.018, 0.047,
## 0.013, 0.077, 0.047, 0.035, 0.046, 0.039, 0.015, 0.026, 0.018,
## 0.01, 0.04, 0.025, 0.047, 0.017, 0.036, 0.019, 0.058, 0.023,
## 0.031, 0.023, 0.048, 0.028, 0, 0.03, 0.054, 0, 0.049, 0.043,
## 0.039, 0.022, 0.02, 0, 0.005, 0.009, 0, 0.01, 0.004, 0.01,
## 0.009, 0.01, 0.014, 0, 0.01, 0.008, 0.007), c(0, 0.006, 0.014,
## 0, 0.01, 0, 0.009, 0.015, 0.008, 0, 0, 0.015, 0, 0.01, 0,
## 0, 0, 0.009, 0, 0, 0, 0.011, 0, 0, 0.007, 0, 0, 0, 0,
## 0, 0.01, 0, 0, 0, 0.007, 0.006, 0, 0, 0, 0, 0.005, 0, 0.007,
## 0.012, 0, 0, 0, 0.011, 0, 0, 0, 0, 0, 0, 0.006, 0, 0.004,
## 0, 0, 0.005, 0, 0.006, 0, 0, 0), c(0.083, 0.106, 0.095, 0.03,
## 0.086, 0.101, 0.045, 0.096, 0.108, 0.073, 0.147, 0.112, 0.088,
## 0.08, 0.054, 0.068, 0.12, 0.102, 0.089, 0.133, 0.095, 0.079,
## 0.104, 0.117, 0.07, 0.112, 0.128, 0.111, 0.118, 0.088, 0.099,
## 0.147, 0.113, 0.066, 0.034, 0.044, 0.153, 0.088, 0.1, 0.093,
## 0.113, 0.101, 0.057, 0.099, 0.036, 0.116, 0.105, 0.135, 0.094,
## 0.05, 0.087, 0.054, 0.09, 0.09, 0.09, 0.119, 0.082, 0.088,
## 0.083, 0.077, 0.116, 0.034, 0.045, 0.104, 0.099, 0.076),
## c(0.575, 0.481, 0.555, 0.646, 0.557, 0.628, 0.521, 0.39,
## 0.579, 0.753, 0.649, 0.674, 0.6, 0.63, 0.597, 0.718, 0.54,
## 0.682, 0.433, 0.576, 0.684, 0.652, 0.654, 0.561, 0.372, 0.573,
## 0.591, 0.599, 0.546, 0.724, 0.745, 0.505, 0.494, 0.564, 0.645,
## 0.706, 0.521, 0.511, 0.68, 0.623, 0.518, 0.609, 0.583, 0.55,
## 0.609, 0.606, 0.776, 0.557, 0.479, 0.621, 0.503, 0.482, 0.483,
## 0.433, 0.523, 0.516, 0.557, 0.495, 0.47, 0.431, 0.399, 0.442,
## 0.469, 0.333, 0.41, 0.41), c(0, 0, 0, 0.015, 0.005, 0, 0,
## 0, 0.004, 0, 0.029, 0.03, 0.032, 0, 0, 0, 0, 0.009, 0, 0.009,
## 0.014, 0.007, 0, 0, 0, 0, 0, 0, 0.009, 0, 0, 0.006, 0,
## 0, 0.007, 0, 0, 0, 0.008, 0.008, 0, 0, 0.021, 0.006, 0.004,
## 0, 0, 0.004, 0.006, 0, 0.005, 0, 0, 0, 0, 0, 0, 0, 0, 0,
## 0, 0.006, 0, 0.008, 0), c(0.056, 0.035, 0.041, 0.03, 0.048,
## 0.043, 0.045, 0.044, 0.05, 0.057, 0.052, 0.007, 0.038, 0.04,
## 0.027, 0.068, 0.098, 0.102, 0.02, 0.08, 0.068, 0.059, 0.033,
## 0.023, 0.028, 0.053, 0.03, 0.074, 0.072, 0.053, 0.02, 0.061,
## 0.069, 0.024, 0.026, 0.036, 0.083, 0.044, 0.039, 0.062, 0.068,
## 0.039, 0.028, 0.021, 0.036, 0.05, 0.038, 0.052, 0.047, 0.067,
## 0.029, 0, 0, 0.005, 0.018, 0, 0, 0, 0.01, 0, 0, 0, 0, 0,
## 0, 0), c(0, 0, 0, 0, 0.014, 0, 0.054, 0.007, 0.029, 0.032,
## 0.052, 0.03, 0.063, 0.01, 0.009, 0.027, 0.023, 0, 0.03, 0.027,
## 0.014, 0, 0, 0.008, 0.07, 0, 0.039, 0.015, 0.013, 0.009,
## 0.109, 0.04, 0.044, 0.028, 0.043, 0.029, 0.013, 0.015, 0.015,

```

```

## 0.008, 0.015, 0.01, 0.014, 0.007, 0.018, 0.008, 0.01, 0.01,
## 0.025, 0, 0.015, 0.01, 0.021, 0.011, 0.054, 0.011, 0.058,
## 0.013, 0.016, 0.034, 0.02, 0.02, 0.067, 0.021, 0.023, 0.007
## ), c(0, 0.012, 0, 0, 0.038, 0.051, 0.036, 0.015, 0.038, 0.008,
## 0, 0.007, 0.044, 0, 0.018, 0.041, 0.008, 0, 0.02, 0.035,
## 0.007, 0.026, 0.005, 0.016, 0.035, 0.013, 0, 0.007, 0.02,
## 0, 0.02, 0.015, 0.05, 0.033, 0.026, 0.029, 0.019, 0.029,
## 0, 0.016, 0.015, 0.005, 0, 0, 0.012, 0.012, 0.01, 0.005,
## 0.025, 0.011, 0.058, 0, 0.014, 0.016, 0.041, 0.011, 0.091,
## 0.021, 0.016, 0.004, 0.01, 0.027, 0.011, 0.021, 0.023, 0.007
## ), c(0, 0.012, 0.007, 0, 0.014, 0.014, 0.009, 0.015, 0.017,
## 0, 0, 0.007, 0.006, 0.02, 0.018, 0.007, 0.015, 0.026, 0.02,
## 0.035, 0.02, 0.007, 0.022, 0, 0.014, 0.033, 0.02, 0, 0.026,
## 0, 0.01, 0.01, 0.019, 0.028, 0.034, 0.022, 0.013, 0, 0, 0.016,
## 0.06, 0.01, 0, 0, 0.024, 0.008, 0.01, 0.005, 0.025, 0.017,
## 0, 0.025, 0.014, 0.005, 0, 0.006, 0.019, 0.017, 0.005, 0.026,
## 0.01, 0.007, 0.022, 0.01, 0.008, 0), c(0.009, 0.012, 0, 0.045,
## 0.019, 0.014, 0, 0.007, 0.008, 0, 0, 0.015, 0.019, 0.01,
## 0.009, 0.014, 0.008, 0.009, 0, 0, 0.02, 0.013, 0.011, 0.016,
## 0, 0, 0.02, 0.007, 0.007, 0, 0.01, 0.02, 0.006, 0.014, 0.009,
## 0.065, 0.013, 0.073, 0.023, 0.016, 0.03, 0.005, 0.014, 0,
## 0, 0.008, 0, 0.003, 0.018, 0.006, 0, 0.005, 0, 0.005, 0.009,
## 0.006, 0.014, 0.004, 0.01, 0.013, 0, 0, 0, 0.01, 0, 0.014
## ), c(0.158, 0.147, 0.156, 0.165, 0.264, 0.159, 0.216, 0.147,
## 0.15, 0.243, 0.111, 0.112, 0.133, 0.19, 0.127, 0.108, 0.15,
## 0.171, 0.226, 0.195, 0.183, 0.145, 0.148, 0.14, 0.148, 0.105,
## 0.118, 0.178, 0.099, 0.221, 0.149, 0.101, 0.144, 0.128, 0.163,
## 0.145, 0.114, 0.16, 0.131, 0.117, 0.113, 0.159, 0.171, 0.183,
## 0.257, 0.158, 0.172, 0.205, 0.187, 0.233, 0.182, 0.192, 0.276,
## 0.232, 0.117, 0.153, 0.139, 0.151, 0.141, 0.145, 0.152, 0.116,
## 0.128, 0.114, 0.213, 0.188), c(0.074, 0.029, 0.007, 0.045,
## 0.029, 0.036, 0.018, 0.022, 0.013, 0.016, 0.007, 0.007, 0.019,
## 0.01, 0.027, 0.061, 0.038, 0.017, 0, 0.027, 0.02, 0.04, 0.022,
## 0.039, 0.049, 0.046, 0.039, 0.074, 0.053, 0.053, 0.129, 0.025,
## 0.013, 0.076, 0.06, 0.029, 0.051, 0, 0.015, 0.093, 0.023,
## 0.024, 0.057, 0, 0, 0.019, 0, 0.026, 0.025, 0.055, 0.022,
## 0.044, 0.007, 0.016, 0.063, 0, 0.043, 0.034, 0.021, 0.021,
## 0.01, 0.007, 0.028, 0.031, 0.038, 0.035), c(0.222, 0.094,
## 0.074, 0.135, 0.091, 0.043, 0.117, 0.044, 0.046, 0.122, 0.059,
## 0.075, 0.07, 0.34, 0.118, 0.075, 0.075, 0.051, 0.03, 0.08,
## 0.108, 0.119, 0.148, 0.164, 0.042, 0.132, 0.069, 0.207, 0.145,
## 0.026, 0.129, 0.03, 0.006, 0.052, 0.069, 0.058, 0.121, 0.015,
## 0.046, 0.156, 0.075, 0.053, 0.085, 0.078, 0.096, 0.116, 0.067,
## 0.062, 0.137, 0.089, 0.051, 0.147, 0.179, 0.053, 0.054, 0.085,
## 0.024, 0.105, 0.031, 0.081, 0.051, 0.218, 0.223, 0.016, 0.076,
## 0.257), c(0.046, 0.129, 0.122, 0.15, 0.086, 0.094, 0.09,
## 0.11, 0.083, 0.13, 0.037, 0.06, 0.051, 0.06, 0.081, 0.081,
## 0.09, 0.094, 0.039, 0.044, 0.081, 0.072, 0.077, 0.07, 0.077,
## 0.04, 0.069, 0.089, 0.145, 0.106, 0.08, 0.086, 0.075, 0.09,
## 0.112, 0.065, 0.095, 0.058, 0.108, 0.055, 0.045, 0.058, 0.057,
## 0.092, 0.066, 0.054, 0.057, 0.075, 0.05, 0.044, 0.102, 0.049,
## 0.048, 0.084, 0.09, 0.068, 0.072, 0.063, 0.083, 0.068, 0.091,
## 0.102, 0.095, 0.099, 0.061, 0.083), c(0.019, 0.27, 0.149,
## 0.21, 0.062, 0.26, 0.108, 0.088, 0.083, 0.032, 0.184, 0.157,

```

```

##      0.101, 0.03, 0.081, 0.129, 0.165, 0.043, 0.226, 0.133, 0.135,
##      0.105, 0.027, 0.125, 0.042, 0.165, 0.167, 0.155, 0.066, 0.044,
##      0.07, 0.126, 0.313, 0.052, 0.138, 0.175, 0.184, 0.131, 0.209,
##      0.179, 0.24, 0.092, 0.071, 0.191, 0.06, 0.081, 0.105, 0.125,
##      0.065, 0.033, 0.058, 0.029, 0.034, 0.032, 0.063, 0.045, 0.029,
##      0.038, 0.078, 0.051, 0.147, 0.075, 0.162, 0.021, 0.023, 0.083
##      ), c(0.074, 0, 0, 0, 0.01, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
##      0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
##      0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.022, 0, 0,
##      0.034, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), c(1, 1, 1,
##      1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
##      1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
##      1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2,
##      2, 2, 2, 2, 2, 2)), control = list(20, 7, 0, 4, 5, 2, 0,
##      30, 0))
##      n= 66
##
##      CP nsplit rel error
## 1 0.9333333      0 1.00000000
## 2 0.0000000      1 0.06666667
##
## Variable importance
## upon there      on      by      to      and
##   31      21      17      12      12      8
##
## Node number 1: 66 observations,      complexity param=0.9333333
## predicted class=Hamilton expected loss=0.2272727 P(node) =1
## class counts:      51      15
## probabilities: 0.773 0.227
## left son=2 (50 obs) right son=3 (16 obs)
## Primary splits:
##      upon < 0.019 to the right, improve=21.306820, (0 missing)
##      there < 0.0145 to the right, improve=13.855950, (0 missing)
##      on < 0.0825 to the left, improve=12.734760, (0 missing)
##      to < 0.499 to the right, improve= 9.556818, (0 missing)
##      by < 0.1385 to the left, improve= 7.973123, (0 missing)
## Surrogate splits:
##      there < 0.0115 to the right, agree=0.924, adj=0.687, (0 split)
##      on < 0.0825 to the left, agree=0.894, adj=0.562, (0 split)
##      by < 0.1385 to the left, agree=0.848, adj=0.375, (0 split)
##      to < 0.4745 to the right, agree=0.848, adj=0.375, (0 split)
##      and < 0.421 to the left, agree=0.818, adj=0.250, (0 split)
##
## Node number 2: 50 observations
## predicted class=Hamilton expected loss=0 P(node) =0.7575758
## class counts:      50      0
## probabilities: 1.000 0.000
##
## Node number 3: 16 observations
## predicted class=Madison expected loss=0.0625 P(node) =0.2424242
## class counts:      1      15
## probabilities: 0.062 0.937

```

```
papers_predict<- predict(papers_model, newdata = data_test, na.action = na.omit, type ="prob")
head(papers_predict, 5)
```

```
##      Hamilton Madison
## 1      0.0625  0.9375
## 2      0.0625  0.9375
## 3      0.0625  0.9375
## 4      0.0625  0.9375
## 5      0.0625  0.9375
```

```
print(papers_predict)
```

```
##      Hamilton Madison
## 1      0.0625  0.9375
## 2      0.0625  0.9375
## 3      0.0625  0.9375
## 4      0.0625  0.9375
## 5      0.0625  0.9375
## 6      0.0625  0.9375
## 7      0.0625  0.9375
## 8      0.0625  0.9375
## 9      0.0625  0.9375
## 10     0.0625  0.9375
## 11     0.0625  0.9375
```

```
papers_predict_raw <- predict(papers_model, newdata = data_test, na.action = na.omit, type ="raw")
print(papers_predict_raw)
```

```
## [1] Madison Madison Madison Madison Madison Madison Madison Madison Madison
## [10] Madison Madison
## Levels: Hamilton Madison
```

## Decision Tree Second Pass Tuning

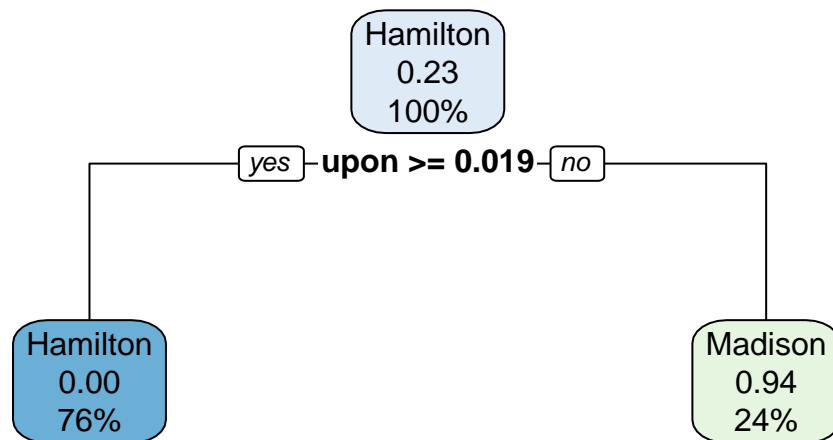
```
data_train <- fedpapers[fedpapers$author == "Hamilton" | fedpapers$author == "Madison",]
data_test <- fedpapers[which(fedpapers$author == "HM" | fedpapers$author == "dispt"),]
```

```
ctrl <- trainControl(method = "cv", number=10)
```

```
papers_model_prob_tuned <- train(author ~., data= data_train, method = "rpart",
                                metric = "Accuracy",
                                tuneGrid=expand.grid(cp=seq(0,0.1,0.01)))
```

```
rpart.plot(papers_model_prob_tuned$finalModel)
```





```
print(papers_model_prob_tuned$finalModel)
```

```
## n= 66
##
## node), split, n, loss, yval, (yprob)
##      * denotes terminal node
##
## 1) root 66 15 Hamilton (0.7727273 0.2272727)
##   2) upon>=0.019 50  0 Hamilton (1.0000000 0.0000000) *
##   3) upon< 0.019 16  1 Madison (0.0625000 0.9375000) *
```

```
summary(papers_model_prob_tuned)
```

```
## Call:
## (function (formula, data, weights, subset, na.action = na.rpart,
##      method, model = FALSE, x = FALSE, y = TRUE, parms, control,
##      cost, ...)
## {
##   Call <- match.call()
##   if (is.data.frame(model)) {
##     m <- model
##     model <- FALSE
##   }
##   else {
```

```

##      indx <- match(c("formula", "data", "weights", "subset"),
##                  names(Call), nomatch = 0)
##      if (indx[1] == 0)
##          stop("a 'formula' argument is required")
##      temp <- Call[c(1, indx)]
##      temp$na.action <- na.action
##      temp[[1]] <- quote(stats::model.frame)
##      m <- eval.parent(temp)
##  }
##  Terms <- attr(m, "terms")
##  if (any(attr(Terms, "order") > 1))
##      stop("Trees cannot handle interaction terms")
##  Y <- model.response(m)
##  wt <- model.weights(m)
##  if (any(wt < 0))
##      stop("negative weights not allowed")
##  if (!length(wt))
##      wt <- rep(1, nrow(m))
##  offset <- model.offset(m)
##  X <- rpart.matrix(m)
##  nobs <- nrow(X)
##  nvar <- ncol(X)
##  if (missing(method)) {
##      method <- if (is.factor(Y) || is.character(Y))
##          "class"
##      else if (inherits(Y, "Surv"))
##          "exp"
##      else if (is.matrix(Y))
##          "poisson"
##      else "anova"
##  }
##  if (is.list(method)) {
##      mlist <- method
##      method <- "user"
##      init <- if (missing(parms))
##          mlist$init(Y, offset, wt = wt)
##      else mlist$init(Y, offset, parms, wt)
##      keep <- rpartcallback(mlist, nobs, init)
##      method.int <- 4
##      parms <- init$parms
##  }
##  else {
##      method.int <- pmatch(method, c("anova", "poisson", "class",
##                                     "exp"))
##      if (is.na(method.int))
##          stop("Invalid method")
##      method <- c("anova", "poisson", "class", "exp")[method.int]
##      if (method.int == 4)
##          method.int <- 2
##      init <- if (missing(parms))
##          get(paste("rpart", method, sep = "."), envir = environment())(Y,
##                                offset, , wt)
##      else get(paste("rpart", method, sep = "."), envir = environment())(Y,
##                                offset, parms, wt)
##  }

```

```

##      ns <- asNamespace("rpart")
##      if (!is.null(init$print))
##        environment(init$print) <- ns
##      if (!is.null(init$summary))
##        environment(init$summary) <- ns
##      if (!is.null(init$text))
##        environment(init$text) <- ns
##    }
##    Y <- init$y
##    xlevels <- .getXlevels(Terms, m)
##    cats <- rep(0, ncol(X))
##    if (!is.null(xlevels))
##      cats[match(names(xlevels), colnames(X))] <- unlist(lapply(xlevels,
##        length))
##    extraArgs <- list(...)
##    if (length(extraArgs)) {
##      controlargs <- names(formals(rpart.control))
##      indx <- match(names(extraArgs), controlargs, nomatch = 0)
##      if (any(indx == 0))
##        stop(gettextf("Argument %s not matched", names(extraArgs)[indx ==
##          0]), domain = NA)
##    }
##    controls <- rpart.control(...)
##    if (!missing(control))
##      controls[names(control)] <- control
##    xval <- controls$xval
##    if (is.null(xval) || (length(xval) == 1 && xval == 0) ||
##      method == "user") {
##      xgroups <- 0
##      xval <- 0
##    }
##    else if (length(xval) == 1) {
##      xgroups <- sample(rep(1:xval, length = nobs), nobs, replace = FALSE)
##    }
##    else if (length(xval) == nobs) {
##      xgroups <- xval
##      xval <- length(unique(xgroups))
##    }
##    else {
##      if (!is.null(attr(m, "na.action"))) {
##        temp <- as.integer(attr(m, "na.action"))
##        xval <- xval[-temp]
##        if (length(xval) == nobs) {
##          xgroups <- xval
##          xval <- length(unique(xgroups))
##        }
##        else stop("Wrong length for 'xval'")
##      }
##      else stop("Wrong length for 'xval'")
##    }
##    if (missing(cost))
##      cost <- rep(1, nvar)
##    else {
##      if (length(cost) != nvar)

```

```

##         stop("Cost vector is the wrong length")
##     if (any(cost <= 0))
##         stop("Cost vector must be positive")
## }
## tfun <- function(x) if (is.matrix(x))
##     rep(is.ordered(x), ncol(x))
## else is.ordered(x)
## labs <- sub("^'(.*)'$", "\\1", attr(Terms, "term.labels"))
## isord <- unlist(lapply(m[labs], tfun))
## storage.mode(X) <- "double"
## storage.mode(wt) <- "double"
## temp <- as.double(unlist(init$parms))
## if (!length(temp))
##     temp <- 0
## rpfit <- .Call(C_rpart, ncat = as.integer(cats * !isord),
##     method = as.integer(method.int), as.double(unlist(controls)),
##     temp, as.integer(xval), as.integer(xgroups), as.double(t(init$y)),
##     X, wt, as.integer(init$numy), as.double(cost))
## nsplit <- nrow(rpfit$split)
## ncat <- if (!is.null(rpfit$csplit))
##     nrow(rpfit$csplit)
## else 0
## if (nsplit == 0)
##     xval <- 0
## numcp <- ncol(rpfit$cptable)
## temp <- if (nrow(rpfit$cptable) == 3)
##     c("CP", "nsplit", "rel error")
## else c("CP", "nsplit", "rel error", "xerror", "xstd")
## dimnames(rpfit$cptable) <- list(temp, 1:numcp)
## tname <- c("<leaf>", colnames(X))
## splits <- matrix(c(rpfit$split[, 2:3], rpfit$dsplit), ncol = 5,
##     dimnames = list(tname[rpfit$split[, 1] + 1], c("count",
##         "ncat", "improve", "index", "adj")))
## index <- rpfit$inode[, 2]
## nadd <- sum(isord[rpfit$split[, 1]])
## if (nadd > 0) {
##     newc <- matrix(0, nadd, max(cats))
##     cvar <- rpfit$split[, 1]
##     indx <- isord[cvar]
##     cdir <- splits[indx, 2]
##     ccut <- floor(splits[indx, 4])
##     splits[indx, 2] <- cats[cvar[indx]]
##     splits[indx, 4] <- ncat + 1:nadd
##     for (i in 1:nadd) {
##         newc[i, 1:(cats[(cvar[indx])[i]])] <- -as.integer(cdir[i])
##         newc[i, 1:ccut[i]] <- as.integer(cdir[i])
##     }
##     catmat <- if (ncat == 0)
##         newc
##     else {
##         cs <- rpfit$csplit
##         ncs <- ncol(cs)
##         ncc <- ncol(newc)
##         if (ncs < ncc)

```

```

##             cs <- cbind(cs, matrix(0, nrow(cs), ncc - ncs))
##             rbind(cs, newc)
##         }
##         ncat <- ncat + nadd
##     }
##     else catmat <- rpfit$csplit
##     if (nsplit == 0) {
##         frame <- data.frame(row.names = 1, var = "<leaf>", n = rpfit$inode[,
##             5], wt = rpfit$dnode[, 3], dev = rpfit$dnode[, 1],
##             yval = rpfit$dnode[, 4], complexity = rpfit$dnode[,
##             2], ncompete = 0, nsurrogate = 0)
##     }
##     else {
##         temp <- ifelse(index == 0, 1, index)
##         svar <- ifelse(index == 0, 0, rpfit$split[temp, 1])
##         frame <- data.frame(row.names = rpfit$inode[, 1], var = tname[svar +
##             1], n = rpfit$inode[, 5], wt = rpfit$dnode[, 3],
##             dev = rpfit$dnode[, 1], yval = rpfit$dnode[, 4],
##             complexity = rpfit$dnode[, 2], ncompete = pmax(0,
##             rpfit$inode[, 3] - 1), nsurrogate = rpfit$inode[,
##             4])
##     }
##     if (method.int == 3) {
##         numclass <- init$numresp - 2
##         nodeprob <- rpfit$dnode[, numclass + 5]/sum(wt)
##         temp <- pmax(1, init$counts)
##         temp <- rpfit$dnode[, 4 + (1:numclass)] %*% diag(init$parms$prior/temp)
##         yprob <- temp/rowSums(temp)
##         yval2 <- matrix(rpfit$dnode[, 4 + (0:numclass)], ncol = numclass +
##             1)
##         frame$yval2 <- cbind(yval2, yprob, nodeprob)
##     }
##     else if (init$numresp > 1)
##         frame$yval2 <- rpfit$dnode[, -(1:3), drop = FALSE]
##     if (is.null(init$summary))
##         stop("Initialization routine is missing the 'summary' function")
##     functions <- if (is.null(init$print))
##         list(summary = init$summary)
##     else list(summary = init$summary, print = init$print)
##     if (!is.null(init$text))
##         functions <- c(functions, list(text = init$text))
##     if (method == "user")
##         functions <- c(functions, mlist)
##     where <- rpfit$which
##     names(where) <- row.names(m)
##     ans <- list(frame = frame, where = where, call = Call, terms = Terms,
##         cptable = t(rpfit$cptable), method = method, parms = init$parms,
##         control = controls, functions = functions, numresp = init$numresp)
##     if (nsplit)
##         ans$splits = splits
##     if (ncat > 0)
##         ans$csplit <- catmat + 2
##     if (nsplit)
##         ans$variable.importance <- importance(ans)

```

```

##   if (model) {
##       ans$model <- m
##       if (missing(y))
##           y <- FALSE
##   }
##   if (y)
##       ans$y <- Y
##   if (x) {
##       ans$x <- X
##       ans$wt <- wt
##   }
##   ans$ordered <- isord
##   if (!is.null(attr(m, "na.action")))
##       ans$na.action <- attr(m, "na.action")
##   if (!is.null(xlevels))
##       attr(ans, "xlevels") <- xlevels
##   if (method == "class")
##       attr(ans, "ylevels") <- init$ylevels
##   class(ans) <- "rpart"
##   ans
## }(formula = .outcome ~ ., data = list(c(0.213, 0.369, 0.305,
## 0.391, 0.327, 0.26, 0.261, 0.449, 0.392, 0.194, 0.361, 0.329,
## 0.329, 0.27, 0.299, 0.325, 0.263, 0.196, 0.344, 0.381, 0.284,
## 0.296, 0.236, 0.382, 0.323, 0.349, 0.315, 0.311, 0.303, 0.203,
## 0.288, 0.323, 0.281, 0.298, 0.378, 0.378, 0.375, 0.379, 0.402,
## 0.358, 0.466, 0.275, 0.27, 0.318, 0.215, 0.32, 0.211, 0.354,
## 0.248, 0.371, 0.328, 0.359, 0.221, 0.29, 0.378, 0.3, 0.25, 0.302,
## 0.24, 0.38, 0.212, 0.136, 0.212, 0.177, 0.243, 0.347), c(0.083,
## 0.07, 0.047, 0.045, 0.096, 0.065, 0.108, 0.022, 0.05, 0.081,
## 0.059, 0.037, 0.051, 0.04, 0.127, 0.027, 0.038, 0.077, 0.059,
## 0.071, 0.041, 0.046, 0.055, 0.023, 0.035, 0.066, 0.02, 0.03,
## 0.053, 0.035, 0.08, 0.086, 0.075, 0.028, 0.017, 0.029, 0.044,
## 0.015, 0.039, 0.023, 0.068, 0.034, 0.028, 0.064, 0.108, 0.042,
## 0.057, 0.068, 0.05, 0.072, 0.058, 0.02, 0.034, 0.048, 0.032,
## 0.074, 0.043, 0.063, 0.042, 0.051, 0.101, 0.054, 0.028, 0.052,
## 0.091, 0.097), c(0, 0.006, 0.007, 0.015, 0, 0, 0, 0, 0.004, 0,
## 0.007, 0.007, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.008, 0, 0, 0.01,
## 0.007, 0.013, 0, 0.02, 0.01, 0, 0, 0.009, 0, 0.006, 0.029, 0.008,
## 0.008, 0.015, 0.005, 0, 0.007, 0.006, 0, 0, 0.005, 0.014, 0.011,
## 0.007, 0, 0.007, 0.016, 0.014, 0.006, 0.014, 0.004, 0, 0.013,
## 0.01, 0.014, 0.006, 0.047, 0.008, 0.007), c(0.083, 0.076, 0.068,
## 0.03, 0.086, 0.087, 0.072, 0.074, 0.075, 0.089, 0.066, 0.052,
## 0.133, 0.05, 0.109, 0.095, 0.053, 0.102, 0.138, 0.071, 0.088,
## 0.086, 0.071, 0.179, 0.07, 0.079, 0.049, 0.081, 0.072, 0.071,
## 0.089, 0.071, 0.094, 0.085, 0.077, 0.073, 0.051, 0.102, 0.077,
## 0.101, 0.143, 0.077, 0.014, 0.085, 0.048, 0.085, 0.096, 0.052,
## 0.054, 0.105, 0.087, 0.069, 0.062, 0.048, 0.072, 0.034, 0.058,
## 0.084, 0.078, 0.051, 0.051, 0.048, 0.05, 0.047, 0.084, 0.056),
##   c(0.343, 0.411, 0.386, 0.27, 0.356, 0.274, 0.467, 0.353,
##   0.329, 0.413, 0.42, 0.329, 0.297, 0.32, 0.308, 0.379, 0.338,
##   0.358, 0.374, 0.372, 0.311, 0.408, 0.33, 0.257, 0.513, 0.217,
##   0.217, 0.274, 0.25, 0.406, 0.258, 0.505, 0.319, 0.36, 0.335,
##   0.364, 0.261, 0.336, 0.247, 0.358, 0.278, 0.338, 0.327, 0.381,
##   0.37, 0.328, 0.374, 0.302, 0.292, 0.388, 0.313, 0.59, 0.407,

```

```

## 0.523, 0.401, 0.346, 0.461, 0.369, 0.491, 0.333, 0.44, 0.422,
## 0.391, 0.436, 0.372, 0.313), c(0.056, 0.023, 0.047, 0.045,
## 0.014, 0.079, 0.018, 0.044, 0.029, 0.065, 0.044, 0.03, 0.063,
## 0.07, 0.018, 0.041, 0.06, 0.043, 0.049, 0.062, 0.047, 0.053,
## 0.055, 0.086, 0, 0.079, 0.049, 0.022, 0.046, 0.018, 0.08,
## 0.03, 0.044, 0.066, 0.06, 0.015, 0.07, 0.044, 0.015, 0.039,
## 0.045, 0.043, 0.114, 0.014, 0.036, 0.058, 0.01, 0.042, 0.108,
## 0.067, 0.029, 0.02, 0.021, 0.011, 0.041, 0.023, 0.01, 0.025,
## 0.057, 0.034, 0.076, 0.027, 0.033, 0.026, 0.008, 0.035),
## c(0.111, 0.053, 0.102, 0.06, 0.086, 0.022, 0.045, 0.059,
## 0.075, 0.138, 0.074, 0.105, 0.038, 0.1, 0.054, 0.061, 0.053,
## 0.154, 0.02, 0.071, 0.129, 0.099, 0.088, 0.047, 0.105, 0.066,
## 0.089, 0.059, 0.026, 0.062, 0.04, 0.025, 0.044, 0.099, 0.043,
## 0.051, 0.051, 0.044, 0.054, 0.023, 0.045, 0.063, 0.085, 0.099,
## 0.119, 0.058, 0.105, 0.112, 0.115, 0.072, 0.102, 0.133, 0.048,
## 0.095, 0.081, 0.113, 0.062, 0.063, 0.047, 0.064, 0.066, 0.048,
## 0.073, 0.135, 0.046, 0.049), c(0.093, 0.117, 0.108, 0.09,
## 0.072, 0.13, 0.027, 0.133, 0.104, 0.186, 0.088, 0.165, 0.088,
## 0.08, 0.091, 0.176, 0.165, 0.128, 0.03, 0.115, 0.122, 0.145,
## 0.154, 0.117, 0.133, 0.125, 0.167, 0.111, 0.132, 0.132, 0.169,
## 0.081, 0.119, 0.099, 0.138, 0.109, 0.07, 0.117, 0.124, 0.117,
## 0.113, 0.13, 0.057, 0.113, 0.108, 0.177, 0.144, 0.127, 0.119,
## 0.111, 0.138, 0.093, 0.159, 0.148, 0.126, 0.187, 0.144, 0.13,
## 0.115, 0.154, 0.136, 0.15, 0.117, 0.083, 0.137, 0.132), c(0.065,
## 0.065, 0.088, 0.015, 0.115, 0.079, 0.063, 0.029, 0.05, 0.024,
## 0.052, 0.082, 0.063, 0.04, 0.063, 0.068, 0.038, 0.043, 0.02,
## 0.018, 0.041, 0.026, 0.016, 0.062, 0.042, 0.053, 0.118, 0.044,
## 0.007, 0.035, 0.06, 0.03, 0.063, 0.057, 0.069, 0.087, 0.064,
## 0.029, 0.015, 0.031, 0.045, 0.014, 0.071, 0.078, 0.018, 0.027,
## 0.019, 0.047, 0.036, 0.033, 0.073, 0.039, 0.048, 0.011, 0.027,
## 0.04, 0.034, 0.021, 0.016, 0.026, 0.02, 0.027, 0.033, 0.036,
## 0.03, 0.035), c(0.315, 0.258, 0.271, 0.376, 0.211, 0.397,
## 0.216, 0.295, 0.221, 0.356, 0.383, 0.389, 0.335, 0.34, 0.371,
## 0.339, 0.36, 0.307, 0.295, 0.284, 0.284, 0.296, 0.434, 0.343,
## 0.126, 0.441, 0.365, 0.481, 0.257, 0.141, 0.358, 0.258, 0.288,
## 0.18, 0.327, 0.298, 0.305, 0.35, 0.216, 0.452, 0.286, 0.319,
## 0.427, 0.247, 0.287, 0.328, 0.326, 0.273, 0.338, 0.188, 0.189,
## 0.3, 0.31, 0.248, 0.23, 0.295, 0.259, 0.311, 0.25, 0.333,
## 0.273, 0.354, 0.402, 0.182, 0.235, 0.361), c(0.028, 0.018,
## 0.054, 0.03, 0.067, 0.051, 0.027, 0.081, 0.067, 0.049, 0.029,
## 0.045, 0.051, 0.06, 0.036, 0.041, 0.075, 0.043, 0.039, 0.098,
## 0.081, 0.033, 0.071, 0.117, 0.084, 0.013, 0.089, 0.089, 0.066,
## 0.15, 0.05, 0.04, 0.05, 0.071, 0.043, 0.029, 0.089, 0.029,
## 0.077, 0.062, 0.068, 0.043, 0.071, 0.113, 0.084, 0.085, 0.01,
## 0.094, 0.065, 0.094, 0.109, 0.044, 0.062, 0.127, 0.068, 0.045,
## 0.096, 0.084, 0.099, 0.06, 0.096, 0.041, 0.028, 0.073, 0.091,
## 0.097), c(0, 0.023, 0.041, 0.03, 0.034, 0.036, 0.009, 0.015,
## 0.05, 0.024, 0.037, 0.022, 0.038, 0.04, 0.027, 0.027, 0.045,
## 0.026, 0.089, 0.08, 0.034, 0.033, 0.033, 0.039, 0.014, 0.02,
## 0, 0.015, 0.026, 0.035, 0.05, 0.025, 0.006, 0.024, 0.026,
## 0.022, 0.032, 0, 0.031, 0.016, 0.023, 0.058, 0.014, 0.056,
## 0.012, 0.042, 0.019, 0.042, 0.047, 0.022, 0.029, 0.044, 0.021,
## 0.048, 0.077, 0.057, 0.019, 0.025, 0.026, 0.03, 0.035, 0.02,
## 0.045, 0.026, 0.023, 0.035), c(0.13, 0.106, 0.095, 0.075,

```

```

## 0.154, 0.094, 0.081, 0.162, 0.125, 0.089, 0.103, 0.165, 0.133,
## 0.14, 0.063, 0.047, 0.105, 0.077, 0.098, 0.124, 0.027, 0.066,
## 0.11, 0.132, 0.077, 0.138, 0.049, 0.118, 0.086, 0.088, 0.119,
## 0.111, 0.175, 0.071, 0.129, 0.116, 0.172, 0.029, 0.124, 0.156,
## 0.12, 0.116, 0.057, 0.078, 0.072, 0.123, 0.038, 0.213, 0.108,
## 0.055, 0.095, 0.177, 0.124, 0.158, 0.162, 0.187, 0.264, 0.139,
## 0.162, 0.205, 0.142, 0.075, 0.112, 0.218, 0.213, 0.153),
## c(0.028, 0.029, 0.014, 0.06, 0.067, 0.007, 0.018, 0.066,
## 0.025, 0.089, 0.022, 0.022, 0.019, 0.11, 0.109, 0.075, 0.083,
## 0.017, 0.01, 0.018, 0.047, 0.04, 0.088, 0.047, 0.014, 0.026,
## 0.049, 0.037, 0.02, 0.026, 0.01, 0.091, 0.019, 0.024, 0.017,
## 0.007, 0.019, 0, 0, 0.008, 0.06, 0.087, 0.1, 0.007, 0.03,
## 0.023, 0.048, 0.016, 0.04, 0.055, 0.007, 0.02, 0.055, 0.011,
## 0.041, 0.006, 0.014, 0.038, 0.016, 0.038, 0.035, 0, 0.039,
## 0.047, 0.03, 0.035), c(0.009, 0.012, 0, 0.015, 0.019, 0.014,
## 0, 0.015, 0.008, 0, 0.015, 0.007, 0, 0, 0.009, 0, 0, 0.009,
## 0, 0.018, 0, 0.007, 0.005, 0.008, 0, 0, 0.01, 0, 0.007, 0,
## 0.01, 0.015, 0, 0.009, 0.009, 0, 0.013, 0, 0, 0.008, 0.008,
## 0.014, 0.014, 0, 0.006, 0.004, 0.01, 0.01, 0.014, 0.006,
## 0, 0, 0, 0.005, 0.005, 0, 0, 0.008, 0.005, 0.009, 0.01, 0,
## 0.006, 0.005, 0, 0.007), c(0, 0, 0, 0, 0.005, 0.007, 0, 0,
## 0, 0, 0, 0, 0, 0.01, 0, 0, 0, 0, 0, 0.007, 0.013, 0, 0,
## 0, 0, 0, 0.007, 0, 0, 0, 0, 0, 0, 0.017, 0, 0, 0, 0.008,
## 0, 0.005, 0, 0, 0.012, 0, 0.01, 0, 0, 0, 0, 0, 0, 0, 0,
## 0, 0.004, 0, 0, 0, 0, 0, 0.01, 0, 0), c(0.019, 0.018, 0.014,
## 0, 0.014, 0.029, 0, 0.015, 0.021, 0, 0.037, 0.022, 0.032,
## 0, 0.018, 0.034, 0.008, 0.017, 0.01, 0, 0.014, 0.013, 0.027,
## 0, 0, 0, 0, 0, 0.02, 0.035, 0.01, 0.015, 0.019, 0.009, 0.009,
## 0.036, 0.013, 0, 0, 0.016, 0, 0.01, 0, 0.007, 0.018, 0.012,
## 0.01, 0.008, 0.011, 0, 0.007, 0, 0.014, 0.011, 0.005, 0,
## 0.024, 0.017, 0.005, 0.013, 0.005, 0.007, 0.011, 0.021, 0,
## 0.014), c(0.028, 0.012, 0.027, 0.045, 0.038, 0.007, 0.009,
## 0.022, 0.029, 0.024, 0.007, 0.007, 0.025, 0, 0, 0, 0.053,
## 0.034, 0.02, 0.035, 0.014, 0.026, 0.011, 0.047, 0.007, 0.013,
## 0.01, 0.022, 0.02, 0, 0.02, 0.02, 0.006, 0.009, 0.026, 0.022,
## 0.025, 0.029, 0.008, 0.031, 0, 0.034, 0.028, 0.021, 0.012,
## 0.035, 0.038, 0.029, 0.022, 0.039, 0.015, 0.01, 0.007, 0.016,
## 0.009, 0.034, 0.038, 0.055, 0.037, 0.038, 0.066, 0.034, 0.022,
## 0.021, 0.023, 0.056), c(0.093, 0.106, 0.054, 0.03, 0.086,
## 0.065, 0.099, 0.037, 0.088, 0.13, 0.074, 0.075, 0.133, 0.07,
## 0.127, 0.115, 0.173, 0.051, 0.039, 0.08, 0.176, 0.072, 0.066,
## 0.101, 0.084, 0.079, 0.167, 0.118, 0.112, 0.071, 0.149, 0.091,
## 0.081, 0.114, 0.086, 0.102, 0.083, 0.131, 0.054, 0.062, 0.143,
## 0.068, 0.213, 0.064, 0.072, 0.081, 0.038, 0.088, 0.09, 0.061,
## 0.073, 0.088, 0.138, 0.063, 0.095, 0.079, 0.111, 0.143, 0.11,
## 0.119, 0.101, 0.082, 0.056, 0.036, 0.114, 0.083), c(0.102,
## 0.111, 0.129, 0.075, 0.101, 0.079, 0.045, 0.118, 0.096, 0.049,
## 0.074, 0.09, 0.063, 0.05, 0.054, 0.088, 0.053, 0.085, 0.089,
## 0.08, 0.081, 0.053, 0.121, 0.093, 0.098, 0.079, 0.079, 0.089,
## 0.072, 0.044, 0.109, 0.061, 0.069, 0.081, 0.12, 0.102, 0.076,
## 0.044, 0.077, 0.078, 0.098, 0.13, 0.1, 0.064, 0.036, 0.073,
## 0.125, 0.049, 0.065, 0.05, 0.044, 0.049, 0.076, 0.1, 0.086,
## 0.125, 0.062, 0.092, 0.052, 0.026, 0.045, 0.054, 0.05, 0.057,
## 0.091, 0.063), c(0.009, 0.006, 0.02, 0, 0.01, 0.036, 0.018,

```



```

## 0.007, 0.013, 0.008, 0.037, 0.03, 0.044, 0, 0.027, 0.014,
## 0, 0, 0.02, 0.027, 0.014, 0.013, 0.016, 0.016, 0.028, 0,
## 0.03, 0.022, 0.013, 0.026, 0, 0.005, 0.044, 0.024, 0.034,
## 0.051, 0.013, 0.044, 0.008, 0.016, 0.023, 0.014, 0, 0.021,
## 0.006, 0.015, 0, 0.013, 0.007, 0.011, 0.036, 0.005, 0.014,
## 0.016, 0.027, 0, 0.067, 0.017, 0.021, 0.013, 0.02, 0.014,
## 0.017, 0.016, 0.076, 0.014), c(0.056, 0.047, 0.088, 0.03,
## 0.062, 0.051, 0.036, 0.066, 0.083, 0.041, 0.066, 0.037, 0.038,
## 0.02, 0.045, 0.034, 0.068, 0.009, 0.02, 0.027, 0.041, 0.02,
## 0.049, 0.047, 0.049, 0.02, 0, 0.007, 0.072, 0.106, 0.06,
## 0.061, 0.025, 0.047, 0.043, 0.036, 0.089, 0.029, 0.054, 0.039,
## 0.06, 0.029, 0.114, 0.056, 0.036, 0.046, 0, 0.057, 0.068,
## 0.083, 0.087, 0.02, 0.055, 0.053, 0.054, 0.051, 0.048, 0.076,
## 0.068, 0.047, 0.035, 0.027, 0.022, 0.088, 0.046, 0.076),
## c(0.093, 0.088, 0.088, 0.06, 0.144, 0.087, 0.036, 0.066,
## 0.058, 0.081, 0.088, 0.082, 0.101, 0.05, 0.1, 0.047, 0.083,
## 0.094, 0.049, 0.115, 0.108, 0.059, 0.121, 0.171, 0.169, 0.04,
## 0.118, 0.126, 0.086, 0.185, 0.139, 0.086, 0.144, 0.137, 0.077,
## 0.051, 0.089, 0.073, 0.124, 0.125, 0.098, 0.092, 0.085, 0.127,
## 0.114, 0.089, 0.067, 0.14, 0.094, 0.127, 0.175, 0.079, 0.11,
## 0.137, 0.095, 0.017, 0.125, 0.109, 0.104, 0.09, 0.167, 0.116,
## 0.084, 0.073, 0.076, 0.125), c(0, 0.076, 0.007, 0.15, 0,
## 0, 0, 0.007, 0.004, 0, 0.007, 0.022, 0, 0, 0.027, 0, 0, 0,
## 0, 0, 0, 0.007, 0, 0, 0.014, 0, 0, 0, 0, 0, 0, 0, 0.019,
## 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.007, 0, 0, 0, 0.013, 0, 0,
## 0, 0, 0.021, 0.005, 0.032, 0, 0, 0.055, 0, 0.004, 0, 0, 0,
## 0, 0, 0), c(0, 0.006, 0.014, 0, 0.005, 0, 0.063, 0.015, 0.029,
## 0, 0.037, 0, 0.038, 0.03, 0, 0.014, 0.015, 0, 0, 0, 0.066,
## 0, 0, 0.021, 0, 0.01, 0.081, 0, 0.044, 0.06, 0.081, 0, 0.062,
## 0.163, 0.247, 0.159, 0.015, 0.062, 0.125, 0.045, 0.01, 0.014,
## 0, 0.012, 0, 0, 0.003, 0.014, 0.039, 0.029, 0.039, 0, 0.026,
## 0.081, 0.006, 0.014, 0.013, 0.005, 0, 0.01, 0.007, 0, 0.052,
## 0.023, 0.007), c(0.028, 0.029, 0.027, 0.045, 0.014, 0.058,
## 0.018, 0.044, 0.025, 0.016, 0.044, 0, 0.082, 0.03, 0.036,
## 0.041, 0.023, 0.017, 0.049, 0.027, 0.034, 0.033, 0.022, 0.023,
## 0.021, 0.02, 0.02, 0.007, 0.079, 0.026, 0.04, 0.035, 0.056,
## 0.028, 0.06, 0.029, 0.013, 0.029, 0.008, 0.008, 0.015, 0.029,
## 0.028, 0.028, 0.018, 0.019, 0.01, 0.049, 0.022, 0.011, 0.007,
## 0.02, 0.014, 0.005, 0.023, 0.011, 0.034, 0.038, 0.016, 0.021,
## 0.03, 0.034, 0.017, 0.016, 0.008, 0.035), c(0.232, 0.387,
## 0.311, 0.24, 0.317, 0.274, 0.252, 0.331, 0.321, 0.211, 0.406,
## 0.285, 0.398, 0.31, 0.353, 0.257, 0.33, 0.401, 0.384, 0.284,
## 0.339, 0.296, 0.341, 0.475, 0.372, 0.494, 0.463, 0.355, 0.454,
## 0.291, 0.398, 0.323, 0.244, 0.374, 0.318, 0.255, 0.375, 0.336,
## 0.34, 0.35, 0.413, 0.309, 0.47, 0.282, 0.269, 0.494, 0.364,
## 0.499, 0.306, 0.394, 0.262, 0.29, 0.269, 0.311, 0.27, 0.351,
## 0.298, 0.248, 0.303, 0.213, 0.298, 0.252, 0.251, 0.27, 0.304,
## 0.368), c(0.019, 0.023, 0.047, 0.105, 0.019, 0.022, 0.027,
## 0, 0.017, 0.016, 0.007, 0.007, 0.051, 0.03, 0.009, 0.027,
## 0.045, 0.043, 0.01, 0.035, 0.014, 0.033, 0.022, 0.039, 0.049,
## 0.033, 0.03, 0.022, 0.013, 0.009, 0, 0.04, 0.013, 0.014,
## 0, 0, 0.025, 0.015, 0.015, 0, 0, 0.01, 0.014, 0.035, 0.006,
## 0.008, 0.01, 0.005, 0.007, 0.011, 0.036, 0.044, 0.014, 0.026,
## 0.032, 0.017, 0.034, 0.029, 0.042, 0.03, 0.056, 0.02, 0.017,

```

```

## 0.005, 0.023, 0.014), c(0.12, 0.053, 0.149, 0.135, 0.207,
## 0.101, 0.099, 0.199, 0.192, 0.154, 0.074, 0.127, 0.139, 0.14,
## 0.118, 0.203, 0.113, 0.196, 0.157, 0.231, 0.108, 0.263, 0.187,
## 0.132, 0.028, 0.178, 0.099, 0.111, 0.191, 0.159, 0.189, 0.258,
## 0.081, 0.232, 0.12, 0.175, 0.184, 0.16, 0.131, 0.156, 0.158,
## 0.237, 0.157, 0.148, 0.137, 0.235, 0.144, 0.299, 0.238, 0.211,
## 0.117, 0.201, 0.166, 0.158, 0.216, 0.323, 0.139, 0.143, 0.204,
## 0.141, 0.142, 0.15, 0.039, 0.218, 0.213, 0.118), c(0.139,
## 0.088, 0.142, 0.09, 0.096, 0.188, 0.09, 0.103, 0.146, 0.154,
## 0.133, 0.15, 0.209, 0.15, 0.145, 0.23, 0.173, 0.179, 0.197,
## 0.284, 0.129, 0.151, 0.176, 0.195, 0.077, 0.25, 0.217, 0.2,
## 0.197, 0.15, 0.129, 0.131, 0.075, 0.152, 0.206, 0.145, 0.178,
## 0.175, 0.162, 0.14, 0.18, 0.193, 0.142, 0.12, 0.137, 0.208,
## 0.086, 0.213, 0.176, 0.188, 0.124, 0.197, 0.179, 0.158, 0.198,
## 0.215, 0.144, 0.164, 0.12, 0.196, 0.081, 0.116, 0.151, 0.088,
## 0.129, 0.111), c(0.074, 0.035, 0.047, 0.03, 0.038, 0.116,
## 0.054, 0.081, 0.067, 0.089, 0.029, 0.067, 0.025, 0.15, 0.072,
## 0.041, 0.15, 0.068, 0.03, 0.115, 0.074, 0.02, 0.093, 0.093,
## 0, 0.066, 0.03, 0.022, 0.026, 0.026, 0.03, 0.035, 0.013,
## 0.038, 0.026, 0.051, 0.057, 0.044, 0.046, 0.047, 0.038, 0.058,
## 0.028, 0.071, 0.048, 0.035, 0.038, 0.055, 0.025, 0.033, 0.036,
## 0.044, 0.048, 0.037, 0.027, 0.113, 0.019, 0.063, 0.047, 0.064,
## 0.03, 0.041, 0.033, 0.052, 0.061, 0.042), c(0.102, 0.053,
## 0.02, 0.12, 0.034, 0.036, 0.036, 0.081, 0.083, 0.049, 0.029,
## 0.022, 0.063, 0.04, 0.063, 0.047, 0.068, 0.085, 0.02, 0.089,
## 0.041, 0.053, 0.082, 0.132, 0.021, 0.066, 0.108, 0.052, 0.053,
## 0.026, 0.05, 0.051, 0.019, 0.076, 0.086, 0.044, 0.076, 0.102,
## 0.031, 0.055, 0.015, 0.077, 0.085, 0.056, 0.084, 0.096, 0.134,
## 0.042, 0.094, 0.089, 0.036, 0.079, 0.11, 0.095, 0.041, 0.034,
## 0.024, 0.097, 0.089, 0.094, 0.03, 0.061, 0.073, 0.042, 0.046,
## 0.069), c(0.065, 0.053, 0.034, 0.09, 0.029, 0.043, 0.081,
## 0.022, 0.025, 0.016, 0.007, 0.015, 0.019, 0.08, 0.081, 0.014,
## 0.023, 0.009, 0, 0.009, 0.027, 0.059, 0.055, 0.07, 0.056,
## 0.026, 0.039, 0.052, 0.079, 0.026, 0.02, 0.035, 0.038, 0.081,
## 0.052, 0.073, 0.064, 0.015, 0.031, 0.039, 0.053, 0.019, 0.071,
## 0.056, 0.012, 0.015, 0.01, 0.06, 0.018, 0.028, 0.044, 0.059,
## 0.055, 0.037, 0.045, 0.034, 0.019, 0.05, 0.005, 0.034, 0.051,
## 0.095, 0.061, 0.047, 0.061, 0.09), c(0.028, 0.012, 0.108,
## 0.045, 0.043, 0.065, 0.009, 0.029, 0.038, 0.073, 0.044, 0.037,
## 0.025, 0.02, 0.063, 0.007, 0.09, 0.068, 0.01, 0.062, 0.027,
## 0.007, 0.055, 0.016, 0.007, 0.04, 0.02, 0.015, 0.033, 0.026,
## 0.01, 0.025, 0.019, 0.043, 0.026, 0.044, 0.006, 0, 0.023,
## 0.039, 0.03, 0.068, 0.028, 0.021, 0.006, 0.012, 0.019, 0.039,
## 0.043, 0.089, 0.007, 0.049, 0.014, 0.111, 0.018, 0.028, 0.014,
## 0.063, 0.021, 0.034, 0.03, 0.041, 0.05, 0.005, 0.008, 0.021
## ), c(0.056, 0, 0.007, 0, 0.014, 0, 0.009, 0, 0.004, 0, 0,
## 0, 0, 0, 0, 0.007, 0, 0, 0, 0, 0.007, 0, 0.011, 0, 0, 0,
## 0.01, 0, 0.007, 0, 0, 0, 0, 0.005, 0, 0, 0, 0, 0.008, 0,
## 0.008, 0, 0, 0, 0, 0.008, 0, 0.013, 0.004, 0.039, 0, 0, 0.014,
## 0, 0, 0, 0, 0, 0, 0, 0, 0.007, 0, 0, 0, 0.007), c(0.028,
## 0.029, 0.007, 0.045, 0.029, 0.029, 0, 0.081, 0.021, 0.016,
## 0.015, 0.015, 0.025, 0.03, 0.054, 0.034, 0.038, 0.026, 0.059,
## 0, 0.047, 0.033, 0.022, 0.062, 0, 0.026, 0.039, 0.022, 0.033,
## 0.026, 0.02, 0.051, 0.019, 0.043, 0.009, 0.044, 0.032, 0,

```

```

## 0.015, 0.031, 0.015, 0.082, 0.043, 0.021, 0.03, 0.031, 0.029,
## 0.055, 0.083, 0.067, 0.015, 0.02, 0.062, 0.026, 0.054, 0.04,
## 0.038, 0.025, 0.042, 0.06, 0.066, 0.041, 0.022, 0.073, 0.038,
## 0.035), c(0.13, 0.117, 0.041, 0.075, 0.077, 0.137, 0.036,
## 0.059, 0.104, 0.081, 0.088, 0.06, 0.133, 0.02, 0.072, 0.061,
## 0.053, 0.102, 0.148, 0.106, 0.095, 0.125, 0.077, 0.101, 0.112,
## 0.086, 0.079, 0.133, 0.138, 0.097, 0.109, 0.056, 0.088, 0.071,
## 0.06, 0.095, 0.076, 0.058, 0.093, 0.109, 0.045, 0.087, 0.114,
## 0.085, 0.09, 0.096, 0.125, 0.091, 0.108, 0.111, 0.095, 0.069,
## 0.09, 0.095, 0.144, 0.125, 0.091, 0.071, 0.115, 0.111, 0.076,
## 0.068, 0.112, 0.062, 0.099, 0.097), c(0, 0.006, 0.007, 0,
## 0.01, 0.007, 0, 0.022, 0.004, 0.008, 0.022, 0, 0, 0, 0, 0,
## 0, 0, 0.02, 0, 0.007, 0.007, 0.005, 0, 0.007, 0.007, 0.01,
## 0, 0, 0.009, 0, 0.005, 0, 0.009, 0, 0, 0.006, 0, 0.015, 0,
## 0.008, 0.014, 0, 0.007, 0.006, 0.023, 0.01, 0.016, 0.025,
## 0.017, 0.015, 0, 0.014, 0, 0.009, 0.011, 0.01, 0.013, 0,
## 0.004, 0.01, 0, 0.006, 0, 0, 0), c(0.908, 1.061, 0.961, 0.887,
## 0.947, 1.04, 0.998, 1.119, 1.008, 0.81, 0.877, 0.936, 0.929,
## 0.94, 0.887, 1.016, 0.833, 0.887, 0.895, 0.807, 0.934, 0.863,
## 0.978, 0.888, 1.075, 0.962, 1.005, 0.969, 1.112, 0.927, 0.885,
## 1.162, 0.982, 0.867, 0.851, 0.88, 0.915, 1.211, 1.152, 0.903,
## 1.089, 0.904, 0.783, 0.96, 0.986, 0.961, 0.9, 0.853, 1.03,
## 0.953, 1.13, 0.772, 0.862, 0.85, 0.856, 0.89, 0.769, 0.856,
## 0.96, 0.951, 0.864, 0.98, 0.91, 0.961, 0.744, 0.806), c(0.074,
## 0.029, 0.081, 0.045, 0.048, 0.029, 0.018, 0.044, 0.038, 0.016,
## 0.066, 0.075, 0.044, 0.04, 0.009, 0.02, 0.06, 0.043, 0.118,
## 0.044, 0.074, 0.053, 0.033, 0.047, 0.014, 0.033, 0.059, 0.037,
## 0.046, 0.026, 0.03, 0.03, 0.088, 0.066, 0.06, 0.044, 0.032,
## 0.015, 0.039, 0.031, 0.023, 0.048, 0.071, 0.078, 0.054, 0.062,
## 0, 0.047, 0.072, 0.1, 0.066, 0.084, 0.11, 0.095, 0.063, 0.125,
## 0.062, 0.097, 0.115, 0.137, 0.147, 0.054, 0.156, 0.099, 0.114,
## 0.118), c(0.028, 0.029, 0.027, 0.135, 0.005, 0.014, 0.018,
## 0.007, 0.038, 0.024, 0.059, 0.015, 0.032, 0.01, 0.036, 0.027,
## 0.038, 0.017, 0.039, 0.018, 0.054, 0.026, 0.033, 0.031, 0.021,
## 0.04, 0.03, 0.044, 0.033, 0.026, 0.05, 0.056, 0.025, 0.038,
## 0.06, 0.022, 0.044, 0.044, 0.031, 0.062, 0.03, 0.034, 0.014,
## 0.042, 0.03, 0.039, 0.067, 0.047, 0.014, 0.044, 0.073, 0.039,
## 0.028, 0.032, 0.054, 0.074, 0.038, 0.034, 0.037, 0.055, 0.025,
## 0.027, 0.045, 0.073, 0.053, 0.028), c(0, 0.029, 0, 0, 0.038,
## 0.022, 0, 0.015, 0.004, 0, 0.022, 0.015, 0.032, 0.04, 0.018,
## 0.027, 0.008, 0.026, 0.039, 0.044, 0.02, 0.013, 0.011, 0.016,
## 0.007, 0.007, 0.039, 0.007, 0.02, 0.044, 0.02, 0.035, 0.006,
## 0.019, 0.017, 0.007, 0.019, 0.015, 0.031, 0.023, 0.03, 0.034,
## 0.043, 0.014, 0.006, 0.023, 0.048, 0.023, 0.032, 0.006, 0.022,
## 0.01, 0.028, 0.053, 0.009, 0.034, 0.019, 0.025, 0.037, 0.026,
## 0.02, 0.007, 0.033, 0.01, 0.008, 0.035), c(0.056, 0.07, 0.047,
## 0.045, 0.183, 0.123, 0.117, 0.147, 0.088, 0.049, 0.088, 0.105,
## 0.076, 0.14, 0.072, 0.108, 0.15, 0.111, 0.089, 0.106, 0.074,
## 0.079, 0.154, 0.055, 0.133, 0.105, 0.079, 0.133, 0.079, 0.044,
## 0.06, 0.071, 0.063, 0.104, 0.138, 0.116, 0.153, 0.175, 0.07,
## 0.179, 0.068, 0.092, 0.057, 0.12, 0.131, 0.081, 0.125, 0.068,
## 0.076, 0.067, 0.08, 0.108, 0.076, 0.069, 0.077, 0.079, 0.043,
## 0.088, 0.078, 0.09, 0.182, 0.054, 0.078, 0.088, 0.084, 0.069
## ), c(0.028, 0.199, 0.034, 0, 0.091, 0.007, 0.018, 0.007,

```

```

## 0.05, 0.057, 0.088, 0.075, 0.032, 0, 0.009, 0.041, 0.023,
## 0.017, 0, 0.009, 0.041, 0.02, 0.011, 0, 0.028, 0, 0, 0, 0.013,
## 0.009, 0.01, 0.005, 0.031, 0.005, 0, 0, 0.013, 0, 0.015,
## 0.008, 0, 0.01, 0.028, 0.035, 0.006, 0.015, 0.01, 0.016,
## 0.014, 0.028, 0, 0.039, 0.083, 0.032, 0.009, 0.011, 0.005,
## 0.025, 0.01, 0.004, 0, 0.027, 0, 0, 0.023, 0), c(0.065, 0.018,
## 0.007, 0.045, 0.005, 0.007, 0.009, 0.037, 0.008, 0.008, 0.007,
## 0.037, 0.006, 0, 0, 0.02, 0, 0, 0.039, 0.018, 0.02, 0.026,
## 0.016, 0.016, 0.007, 0.013, 0.02, 0, 0.007, 0.044, 0.04,
## 0.03, 0.006, 0.009, 0.009, 0.007, 0.019, 0.015, 0.046, 0.055,
## 0.023, 0.014, 0.057, 0, 0.03, 0.027, 0.077, 0.013, 0.079,
## 0.017, 0.029, 0.005, 0, 0, 0.014, 0, 0.029, 0.017, 0.047,
## 0.03, 0.076, 0, 0, 0.036, 0.023, 0.021), c(0.009, 0.018,
## 0.027, 0, 0.029, 0.043, 0.018, 0.037, 0.033, 0.016, 0.044,
## 0.037, 0.038, 0.02, 0.072, 0.054, 0.03, 0.051, 0.03, 0.035,
## 0.041, 0.066, 0.027, 0.008, 0.014, 0.033, 0.069, 0.022, 0.033,
## 0.009, 0.05, 0.03, 0.025, 0.028, 0.052, 0, 0.044, 0.029,
## 0, 0.008, 0.015, 0.024, 0.028, 0.035, 0.018, 0.031, 0.048,
## 0.023, 0.004, 0.022, 0.022, 0, 0.014, 0.053, 0.045, 0.006,
## 0.053, 0, 0, 0.038, 0.02, 0.014, 0.05, 0.01, 0.046, 0.007
## ), c(0.019, 0.029, 0.041, 0.045, 0.01, 0.022, 0.036, 0.052,
## 0.038, 0.008, 0.029, 0.007, 0.063, 0.02, 0.009, 0.054, 0.015,
## 0.06, 0, 0.027, 0.02, 0.04, 0.005, 0.023, 0.021, 0.04, 0.02,
## 0.037, 0.039, 0.062, 0.06, 0.01, 0.025, 0.052, 0.043, 0.007,
## 0.032, 0.044, 0.046, 0.016, 0.03, 0.029, 0.014, 0.035, 0.024,
## 0.019, 0.01, 0.018, 0.022, 0.028, 0.022, 0.015, 0.041, 0.042,
## 0.032, 0.028, 0.062, 0.038, 0.042, 0.026, 0.03, 0.007, 0.006,
## 0.021, 0.038, 0.007), c(0, 0.006, 0.014, 0, 0.01, 0.022,
## 0.009, 0.022, 0.017, 0.008, 0.029, 0.037, 0.025, 0.01, 0,
## 0.007, 0.008, 0.026, 0, 0.009, 0.027, 0.007, 0, 0.031, 0.028,
## 0.02, 0.02, 0.03, 0.013, 0.009, 0.01, 0.005, 0.038, 0.005,
## 0.017, 0.022, 0.013, 0, 0.015, 0.031, 0.045, 0.01, 0.028,
## 0.014, 0.048, 0.019, 0.01, 0.021, 0.004, 0.011, 0.007, 0.015,
## 0.048, 0.021, 0.045, 0.011, 0.034, 0.004, 0.005, 0.009, 0.025,
## 0.007, 0.006, 0.016, 0.046, 0.028), c(0.009, 0.029, 0.02,
## 0.015, 0.005, 0.022, 0, 0.022, 0.038, 0.057, 0.029, 0.03,
## 0.057, 0.03, 0.018, 0.027, 0.03, 0.034, 0.049, 0.035, 0.02,
## 0.02, 0.038, 0.023, 0.014, 0.026, 0, 0.03, 0.039, 0.018,
## 0.03, 0.02, 0.019, 0.024, 0.034, 0.051, 0.013, 0.044, 0.015,
## 0.039, 0.03, 0.034, 0.071, 0.035, 0.036, 0.05, 0.038, 0.023,
## 0.029, 0.028, 0.029, 0.039, 0.007, 0.021, 0.05, 0.023, 0.053,
## 0.025, 0.042, 0.051, 0.035, 0.014, 0.033, 0.01, 0, 0.028),
## c(0.102, 0.035, 0.027, 0.15, 0.048, 0.029, 0.045, 0.029,
## 0.046, 0, 0.029, 0.03, 0.032, 0.07, 0.027, 0.014, 0.023,
## 0.051, 0.01, 0.009, 0.02, 0.053, 0.033, 0.055, 0.042, 0.046,
## 0.118, 0.037, 0.046, 0.044, 0.06, 0.061, 0.031, 0.066, 0.043,
## 0.051, 0.089, 0.029, 0.062, 0.055, 0.045, 0.048, 0.1, 0.021,
## 0.024, 0.023, 0.029, 0.042, 0.058, 0.028, 0.007, 0.064, 0.034,
## 0.026, 0.05, 0.045, 0.024, 0.055, 0.021, 0.043, 0.04, 0.061,
## 0.061, 0.026, 0.03, 0.035), c(0.241, 0.141, 0.176, 0.27,
## 0.178, 0.116, 0.189, 0.177, 0.221, 0.227, 0.243, 0.225, 0.265,
## 0.38, 0.172, 0.196, 0.233, 0.256, 0.315, 0.284, 0.284, 0.244,
## 0.225, 0.226, 0.155, 0.303, 0.315, 0.17, 0.237, 0.141, 0.189,
## 0.227, 0.169, 0.152, 0.189, 0.131, 0.248, 0.19, 0.271, 0.265,

```

```

## 0.24, 0.324, 0.256, 0.127, 0.149, 0.185, 0.182, 0.309, 0.302,
## 0.188, 0.182, 0.152, 0.221, 0.243, 0.252, 0.153, 0.298, 0.172,
## 0.167, 0.209, 0.142, 0.204, 0.212, 0.171, 0.205, 0.222),
## c(1.094, 1.044, 1.097, 1.052, 1.163, 1.322, 1.366, 1.23,
## 1.125, 1.45, 1.04, 1.205, 1.195, 1.47, 1.43, 1.436, 1.178,
## 1.356, 1.377, 1.339, 1.232, 1.199, 1.275, 1.309, 1.11, 1.416,
## 1.438, 1.539, 1.56, 1.448, 1.352, 1.415, 1.188, 1.303, 1.487,
## 1.258, 1.246, 1.43, 1.515, 1.199, 1.27, 1.435, 1.11, 0.953,
## 1.368, 1.404, 1.465, 1.165, 1.303, 1.275, 1.151, 1.204, 1.352,
## 1.135, 1.18, 1.552, 1.259, 1.293, 1.247, 1.395, 1.501, 1.803,
## 1.613, 1.522, 1.184, 1.389), c(0.13, 0.059, 0.095, 0.015,
## 0.034, 0.087, 0.09, 0.059, 0.092, 0.065, 0.096, 0.082, 0.063,
## 0.06, 0.118, 0.095, 0.06, 0.085, 0.039, 0.044, 0.068, 0.125,
## 0.077, 0.078, 0.091, 0.092, 0.049, 0.037, 0.092, 0.026, 0.099,
## 0.03, 0.1, 0.104, 0.12, 0.087, 0.051, 0.029, 0.015, 0.055,
## 0.083, 0.155, 0.171, 0.106, 0.036, 0.042, 0.057, 0.039, 0.086,
## 0.039, 0.102, 0.103, 0.131, 0.121, 0.113, 0.091, 0.168, 0.055,
## 0.063, 0.043, 0.045, 0.048, 0.162, 0.005, 0.099, 0.09), c(0,
## 0.012, 0, 0, 0, 0, 0, 0, 0.004, 0, 0, 0.015, 0.019, 0, 0.018,
## 0, 0.008, 0, 0.02, 0.009, 0.007, 0, 0.005, 0, 0, 0.007, 0.01,
## 0, 0.007, 0.009, 0, 0, 0, 0.005, 0.009, 0.007, 0, 0, 0.008,
## 0.008, 0, 0.01, 0.014, 0, 0.006, 0.004, 0.019, 0.003, 0.004,
## 0.011, 0.007, 0.01, 0, 0.005, 0.005, 0.017, 0.019, 0.008,
## 0.005, 0, 0, 0.014, 0, 0.01, 0, 0), c(0.019, 0.023, 0.041,
## 0.105, 0.072, 0.022, 0.018, 0.029, 0.054, 0.032, 0.037, 0.007,
## 0.051, 0.08, 0.063, 0.068, 0.038, 0.051, 0.059, 0.018, 0.047,
## 0.013, 0.077, 0.047, 0.035, 0.046, 0.039, 0.015, 0.026, 0.018,
## 0.01, 0.04, 0.025, 0.047, 0.017, 0.036, 0.019, 0.058, 0.023,
## 0.031, 0.023, 0.048, 0.028, 0, 0.03, 0.054, 0, 0.049, 0.043,
## 0.039, 0.022, 0.02, 0, 0.005, 0.009, 0, 0.01, 0.004, 0.01,
## 0.009, 0.01, 0.014, 0, 0.01, 0.008, 0.007), c(0, 0.006, 0.014,
## 0, 0.01, 0, 0.009, 0.015, 0.008, 0, 0, 0.015, 0, 0.01, 0,
## 0, 0, 0.009, 0, 0, 0, 0.011, 0, 0, 0.007, 0, 0, 0, 0,
## 0, 0.01, 0, 0, 0, 0.007, 0.006, 0, 0, 0, 0, 0.005, 0, 0.007,
## 0.012, 0, 0, 0, 0.011, 0, 0, 0, 0, 0, 0, 0.006, 0, 0.004,
## 0, 0, 0.005, 0, 0.006, 0, 0, 0), c(0.083, 0.106, 0.095, 0.03,
## 0.086, 0.101, 0.045, 0.096, 0.108, 0.073, 0.147, 0.112, 0.088,
## 0.08, 0.054, 0.068, 0.12, 0.102, 0.089, 0.133, 0.095, 0.079,
## 0.104, 0.117, 0.07, 0.112, 0.128, 0.111, 0.118, 0.088, 0.099,
## 0.147, 0.113, 0.066, 0.034, 0.044, 0.153, 0.088, 0.1, 0.093,
## 0.113, 0.101, 0.057, 0.099, 0.036, 0.116, 0.105, 0.135, 0.094,
## 0.05, 0.087, 0.054, 0.09, 0.09, 0.09, 0.119, 0.082, 0.088,
## 0.083, 0.077, 0.116, 0.034, 0.045, 0.104, 0.099, 0.076),
## c(0.575, 0.481, 0.555, 0.646, 0.557, 0.628, 0.521, 0.39,
## 0.579, 0.753, 0.649, 0.674, 0.6, 0.63, 0.597, 0.718, 0.54,
## 0.682, 0.433, 0.576, 0.684, 0.652, 0.654, 0.561, 0.372, 0.573,
## 0.591, 0.599, 0.546, 0.724, 0.745, 0.505, 0.494, 0.564, 0.645,
## 0.706, 0.521, 0.511, 0.68, 0.623, 0.518, 0.609, 0.583, 0.55,
## 0.609, 0.606, 0.776, 0.557, 0.479, 0.621, 0.503, 0.482, 0.483,
## 0.433, 0.523, 0.516, 0.557, 0.495, 0.47, 0.431, 0.399, 0.442,
## 0.469, 0.333, 0.41, 0.41), c(0, 0, 0, 0.015, 0.005, 0, 0,
## 0, 0.004, 0, 0.029, 0.03, 0.032, 0, 0, 0, 0, 0.009, 0, 0.009,
## 0.014, 0.007, 0, 0, 0, 0, 0, 0, 0.009, 0, 0, 0.006, 0,
## 0, 0.007, 0, 0, 0, 0.008, 0.008, 0, 0, 0.021, 0.006, 0.004,

```

```

## 0, 0, 0.004, 0.006, 0, 0.005, 0, 0, 0, 0, 0, 0, 0, 0,
## 0, 0.006, 0, 0.008, 0), c(0.056, 0.035, 0.041, 0.03, 0.048,
## 0.043, 0.045, 0.044, 0.05, 0.057, 0.052, 0.007, 0.038, 0.04,
## 0.027, 0.068, 0.098, 0.102, 0.02, 0.08, 0.068, 0.059, 0.033,
## 0.023, 0.028, 0.053, 0.03, 0.074, 0.072, 0.053, 0.02, 0.061,
## 0.069, 0.024, 0.026, 0.036, 0.083, 0.044, 0.039, 0.062, 0.068,
## 0.039, 0.028, 0.021, 0.036, 0.05, 0.038, 0.052, 0.047, 0.067,
## 0.029, 0, 0, 0.005, 0.018, 0, 0, 0, 0.01, 0, 0, 0, 0, 0,
## 0, 0), c(0, 0, 0, 0, 0.014, 0, 0.054, 0.007, 0.029, 0.032,
## 0.052, 0.03, 0.063, 0.01, 0.009, 0.027, 0.023, 0, 0.03, 0.027,
## 0.014, 0, 0, 0.008, 0.07, 0, 0.039, 0.015, 0.013, 0.009,
## 0.109, 0.04, 0.044, 0.028, 0.043, 0.029, 0.013, 0.015, 0.015,
## 0.008, 0.015, 0.01, 0.014, 0.007, 0.018, 0.008, 0.01, 0.01,
## 0.025, 0, 0.015, 0.01, 0.021, 0.011, 0.054, 0.011, 0.058,
## 0.013, 0.016, 0.034, 0.02, 0.02, 0.067, 0.021, 0.023, 0.007
## ), c(0, 0.012, 0, 0, 0.038, 0.051, 0.036, 0.015, 0.038, 0.008,
## 0, 0.007, 0.044, 0, 0.018, 0.041, 0.008, 0, 0.02, 0.035,
## 0.007, 0.026, 0.005, 0.016, 0.035, 0.013, 0, 0.007, 0.02,
## 0, 0.02, 0.015, 0.05, 0.033, 0.026, 0.029, 0.019, 0.029,
## 0, 0.016, 0.015, 0.005, 0, 0, 0.012, 0.012, 0.01, 0.005,
## 0.025, 0.011, 0.058, 0, 0.014, 0.016, 0.041, 0.011, 0.091,
## 0.021, 0.016, 0.004, 0.01, 0.027, 0.011, 0.021, 0.023, 0.007
## ), c(0, 0.012, 0.007, 0, 0.014, 0.014, 0.009, 0.015, 0.017,
## 0, 0, 0.007, 0.006, 0.02, 0.018, 0.007, 0.015, 0.026, 0.02,
## 0.035, 0.02, 0.007, 0.022, 0, 0.014, 0.033, 0.02, 0, 0.026,
## 0, 0.01, 0.01, 0.019, 0.028, 0.034, 0.022, 0.013, 0, 0, 0.016,
## 0.06, 0.01, 0, 0, 0.024, 0.008, 0.01, 0.005, 0.025, 0.017,
## 0, 0.025, 0.014, 0.005, 0, 0.006, 0.019, 0.017, 0.005, 0.026,
## 0.01, 0.007, 0.022, 0.01, 0.008, 0), c(0.009, 0.012, 0, 0.045,
## 0.019, 0.014, 0, 0.007, 0.008, 0, 0, 0.015, 0.019, 0.01,
## 0.009, 0.014, 0.008, 0.009, 0, 0, 0.02, 0.013, 0.011, 0.016,
## 0, 0, 0.02, 0.007, 0.007, 0, 0.01, 0.02, 0.006, 0.014, 0.009,
## 0.065, 0.013, 0.073, 0.023, 0.016, 0.03, 0.005, 0.014, 0,
## 0, 0.008, 0, 0.003, 0.018, 0.006, 0, 0.005, 0, 0.005, 0.009,
## 0.006, 0.014, 0.004, 0.01, 0.013, 0, 0, 0, 0.01, 0, 0.014
## ), c(0.158, 0.147, 0.156, 0.165, 0.264, 0.159, 0.216, 0.147,
## 0.15, 0.243, 0.111, 0.112, 0.133, 0.19, 0.127, 0.108, 0.15,
## 0.171, 0.226, 0.195, 0.183, 0.145, 0.148, 0.14, 0.148, 0.105,
## 0.118, 0.178, 0.099, 0.221, 0.149, 0.101, 0.144, 0.128, 0.163,
## 0.145, 0.114, 0.16, 0.131, 0.117, 0.113, 0.159, 0.171, 0.183,
## 0.257, 0.158, 0.172, 0.205, 0.187, 0.233, 0.182, 0.192, 0.276,
## 0.232, 0.117, 0.153, 0.139, 0.151, 0.141, 0.145, 0.152, 0.116,
## 0.128, 0.114, 0.213, 0.188), c(0.074, 0.029, 0.007, 0.045,
## 0.029, 0.036, 0.018, 0.022, 0.013, 0.016, 0.007, 0.007, 0.019,
## 0.01, 0.027, 0.061, 0.038, 0.017, 0, 0.027, 0.02, 0.04, 0.022,
## 0.039, 0.049, 0.046, 0.039, 0.074, 0.053, 0.053, 0.129, 0.025,
## 0.013, 0.076, 0.06, 0.029, 0.051, 0, 0.015, 0.093, 0.023,
## 0.024, 0.057, 0, 0, 0.019, 0, 0.026, 0.025, 0.055, 0.022,
## 0.044, 0.007, 0.016, 0.063, 0, 0.043, 0.034, 0.021, 0.021,
## 0.01, 0.007, 0.028, 0.031, 0.038, 0.035), c(0.222, 0.094,
## 0.074, 0.135, 0.091, 0.043, 0.117, 0.044, 0.046, 0.122, 0.059,
## 0.075, 0.07, 0.34, 0.118, 0.075, 0.075, 0.051, 0.03, 0.08,
## 0.108, 0.119, 0.148, 0.164, 0.042, 0.132, 0.069, 0.207, 0.145,
## 0.026, 0.129, 0.03, 0.006, 0.052, 0.069, 0.058, 0.121, 0.015,

```

```

##      0.046, 0.156, 0.075, 0.053, 0.085, 0.078, 0.096, 0.116, 0.067,
##      0.062, 0.137, 0.089, 0.051, 0.147, 0.179, 0.053, 0.054, 0.085,
##      0.024, 0.105, 0.031, 0.081, 0.051, 0.218, 0.223, 0.016, 0.076,
##      0.257), c(0.046, 0.129, 0.122, 0.15, 0.086, 0.094, 0.09,
##      0.11, 0.083, 0.13, 0.037, 0.06, 0.051, 0.06, 0.081, 0.081,
##      0.09, 0.094, 0.039, 0.044, 0.081, 0.072, 0.077, 0.07, 0.077,
##      0.04, 0.069, 0.089, 0.145, 0.106, 0.08, 0.086, 0.075, 0.09,
##      0.112, 0.065, 0.095, 0.058, 0.108, 0.055, 0.045, 0.058, 0.057,
##      0.092, 0.066, 0.054, 0.057, 0.075, 0.05, 0.044, 0.102, 0.049,
##      0.048, 0.084, 0.09, 0.068, 0.072, 0.063, 0.083, 0.068, 0.091,
##      0.102, 0.095, 0.099, 0.061, 0.083), c(0.019, 0.27, 0.149,
##      0.21, 0.062, 0.26, 0.108, 0.088, 0.083, 0.032, 0.184, 0.157,
##      0.101, 0.03, 0.081, 0.129, 0.165, 0.043, 0.226, 0.133, 0.135,
##      0.105, 0.027, 0.125, 0.042, 0.165, 0.167, 0.155, 0.066, 0.044,
##      0.07, 0.126, 0.313, 0.052, 0.138, 0.175, 0.184, 0.131, 0.209,
##      0.179, 0.24, 0.092, 0.071, 0.191, 0.06, 0.081, 0.105, 0.125,
##      0.065, 0.033, 0.058, 0.029, 0.034, 0.032, 0.063, 0.045, 0.029,
##      0.038, 0.078, 0.051, 0.147, 0.075, 0.162, 0.021, 0.023, 0.083
##      ), c(0.074, 0, 0, 0, 0.01, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
##      0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
##      0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.022, 0, 0,
##      0.034, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), c(1, 1, 1,
##      1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
##      1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
##      1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
##      1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2,
##      2, 2, 2, 2, 2, 2)), control = list(20, 7, 0, 4, 5, 2, 0,
##      30, 0))
##      n= 66
##
##      CP nsplit rel error
## 1 0.9333333      0 1.0000000
## 2 0.0000000      1 0.0666667
##
## Variable importance
## upon there      on      by      to      and
##   31      21      17      12      12      8
##
## Node number 1: 66 observations,      complexity param=0.9333333
## predicted class=Hamilton expected loss=0.2272727 P(node) =1
## class counts:      51      15
## probabilities: 0.773 0.227
## left son=2 (50 obs) right son=3 (16 obs)
## Primary splits:
##      upon < 0.019 to the right, improve=21.306820, (0 missing)
##      there < 0.0145 to the right, improve=13.855950, (0 missing)
##      on < 0.0825 to the left, improve=12.734760, (0 missing)
##      to < 0.499 to the right, improve= 9.556818, (0 missing)
##      by < 0.1385 to the left, improve= 7.973123, (0 missing)
## Surrogate splits:
##      there < 0.0115 to the right, agree=0.924, adj=0.687, (0 split)
##      on < 0.0825 to the left, agree=0.894, adj=0.562, (0 split)
##      by < 0.1385 to the left, agree=0.848, adj=0.375, (0 split)
##      to < 0.4745 to the right, agree=0.848, adj=0.375, (0 split)
##      and < 0.421 to the left, agree=0.818, adj=0.250, (0 split)

```

```
##
## Node number 2: 50 observations
##   predicted class=Hamilton   expected loss=0   P(node) =0.7575758
##   class counts:      50      0
##   probabilities: 1.000 0.000
##
## Node number 3: 16 observations
##   predicted class=Madison   expected loss=0.0625   P(node) =0.2424242
##   class counts:         1      15
##   probabilities: 0.062 0.937
```

```
predict_secondPass_tuned <- predict(papers_model_prob_tuned, newdata = data_test, na.action = na.omit,
print(predict_secondPass_tuned)
```

```
##      Hamilton Madison
## 1      0.0625  0.9375
## 2      0.0625  0.9375
## 3      0.0625  0.9375
## 4      0.0625  0.9375
## 5      0.0625  0.9375
## 6      0.0625  0.9375
## 7      0.0625  0.9375
## 8      0.0625  0.9375
## 9      0.0625  0.9375
## 10     0.0625  0.9375
## 11     0.0625  0.9375
```