Lab 4, Week 6

Sidi Wu

Testing the revised `recpart()` and `recpart_recursive()`

1. The code chunk at the end of the lab includes revised `recpart()` and `recpart_recursive()` functions that include a `debug` option, set to `FALSE` by default. If `TRUE`, `recpart_recursive()` prints the variable and split point of each split it encounters. Run `recpart()` on the test dataset from lab 3 and with `debug=TRUE` to see the split points.

```
new_node <- function(data,childl=NULL,childr=NULL){
  nn <- list(data=data,childl=childl,childr=childr)
  class(nn) <- "node"
  return(nn)
}
new_region <- function(coords=NULL,x,y){
  if(is.null(coords)) {
    coords <- sapply(x,range)
  }
  out <- list(coords=coords,x=x,y=y)
  class(out) <- "region"
  out
}
recpart <- function(x,y,debug=FALSE){
  init <- new_node(new_region(x=x,y=y))
  tree <- recpart_recursive(init,debug)
  class(tree) <- c("tree",class(tree))
  return(tree)
}
recpart_recursive <- function(node,debug=FALSE) {
  R <- node$data
  if(length(R$y) == 1) { return(node) }
  lof_best  <- Inf
  for(v in 1:ncol(R$x)){
    tt <- split_points(R$x[,v])
    for(t in tt) {
      gdat <- data.frame(y=R$y,x=as.numeric(R$x[,v] <= t))
      lof <- LOF(y~.,gdat)
      if(lof < lof_best) {
        lof_best <- lof
        if(debug) best_split <- c(v=v,t=t)
        childRs <- split(R,xvar=v,spt=t)
      }
    }
  }
  if(debug) {
    cat("best split on variable",best_split["v"], "at", best_split["t"],"\n")
```

```
  }
  node$childl <- recpart_recursive(new_node(childRs$Rl),debug)
  node$childr <- recpart_recursive(new_node(childRs$Rr),debug)
  return(node)
}
split_points <- function(x) {
  x <- sort(unique(x))
  x <- x[-length(x)]
  return(x)
}
LOF <- function(form,data) {
  ff <- lm(form,data)
  return(sum(residuals(ff)^2))
}
split.region <- function(R,xvar,spt){
  r1_ind <- (R$x[,xvar] <= spt)
  c1 <- c2 <- R$coords
  c1[2,xvar] <- spt; c2[1,xvar] <- spt
  Rl <- new_region(c1,R$x[r1_ind,,drop=FALSE],R$y[r1_ind])
  Rr <- new_region(c2,R$x[!r1_ind,,drop=FALSE],R$y[!r1_ind])
  return(list(Rl=Rl,Rr=Rr))
}
#-------------------------------------------------#
```