RecursiveBinarySplitting

Justin Goodwater

```r
# build funtion to compute RSS
get_rss <- function(y, X, j, s) {
  R1 = X[ , j] < s
  R2 = X[ , j] >= s
  pred_R1 = mean(y[R1])
  pred_R2 = mean(y[R2])
  (y[R1] - pred_R1)^2
  rss = sum((y[R1] - pred_R1)^2) + sum((y[R2] - pred_R2)^2)
  return(rss)
}
```

```r
# load data and call function above
data = read.csv("tree_ex.csv", header = FALSE)
n = dim(data)[1]
y = data[ , 1]
X = data[ , 2:3]
rss = get_rss(y, X, 2, 0)
store_rss = matrix(0, 100, 1)
```

```r
# build grid for cut points s for predictor j = 1 and compute associated RSS
sgrid <- seq(0, 10, length.out = 100)
for (i in 1:100){
  s = sgrid[i]
  store_rss[i] = get_rss(y, X, 1, s)
}
sgrid[which.min(store_rss)]
```

```
## [1] 6.969697
```

```r
# repeat for predictor j = 2 and compute associated RSS
sgrid <- seq(-5, 5, length.out = 100)
for (i in 1:100){
  s = sgrid[i]
  store_rss[i] = get_rss(y, X, 2, s)
}
sgrid[which.min(store_rss)]
```

```
## [1] 2.979798
```