PS6

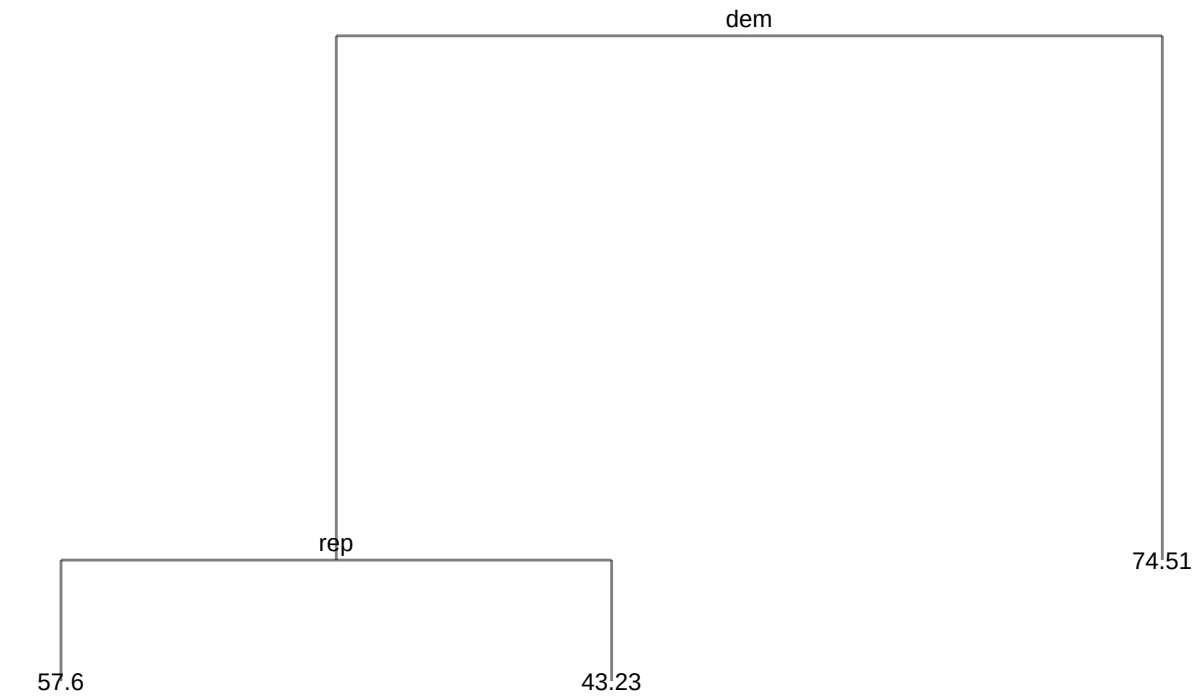
Dan Gamarnik

February 26, 2018

A).

```
set.seed(1234)
biden_split <- resample_partition(biden_data, c(test = .3, train = .7))
```

```
# estimate model
biden_tree <- tree(biden ~ ., data = biden_split$train)

# plot tree
tree_data <- dendro_data(biden_tree)


ptree <- ggplot(segment(tree_data)) +
  geom_segment(aes(x = x, y = y, xend = xend, yend = yend),
               alpha = 0.5) +
  geom_text(data = label(tree_data),
            aes(x = x, y = y, label = label), vjust = -0.5, size = 3) +
  geom_text(data = leaf_label(tree_data),
            aes(x = x, y = y, label = label), vjust = 0.5, size = 3) +
  theme_dendro()+
  labs(title = 'Decision Tree for Biden Scores',
       subtitle = 'All predictors, Default Controls')

# function to get MSE
mse <- function(model, data) {
  x <- modelr:::residuals(model, data)
  mean(x ^ 2, na.rm = TRUE)
}

mse_biden_1 = mse(biden_tree,biden_split$test)
leaf_vals <- leaf_label(tree_data)$yval
ptree
```



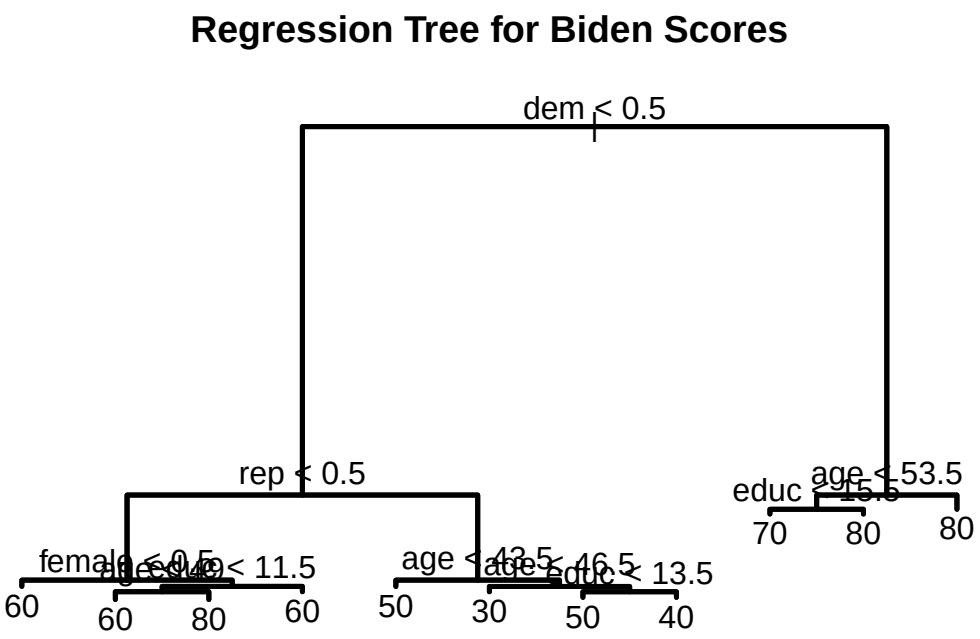
This tree predicts democrats will have a biden score of 74.51. The GOP are predicted to have a bidenscore of 43.23 and respondents who are neither democrat nor republican are predicted to have a biden score of 57.6.

The MSE of this tree is 406.

B).

```
set.seed(1234) # For reproducibility

biden_tree_2 <- tree(biden ~ ., data = biden_split$train,
     control = tree.control(nobs = nrow(biden_split$train),
                            mindev = 0))
mod <- biden_tree_2

mse_biden_2 <- mse(biden_tree_2, biden_split$test)

num_nodes <- 2:25
pruned_trees <- map(num_nodes, prune.tree, tree = biden_tree_2, k = NULL)
test_mses <- map_dbl(pruned_trees, mse, data = biden_split$test)

tree.opt <- pruned_trees[[which.min(test_mses)]]
opt_test_mse <- mse(tree.opt, biden_split$test)


biden_pruned <- prune.tree(biden_tree_2, best=11)
mse_pruned = mse(biden_pruned,biden_split$test)
```

```
plot(biden_pruned, col='black', lwd=2.5)
title("Regression Tree for Biden Scores")
text(biden_pruned, col='black')
```



From this tree we can see that party affiliation continues to be the most important indicator for biden warmth. For the GOP, gender is the second most important indicator. For democrats and unaffiliated, age is the second strongest predictor of warmth score. The MSE of this tree is 401.

C).

```
df = read.csv('biden.csv')
df$Party[df$dem == 1] = 'Democrat'
df$Party[df$dem == 0 & df$rep == 0] = 'No Affiliation'
df$Party[df$rep == 1] = 'Republican'

set.seed(1234)

biden_split7030 = resample_partition(df, c(test = 0.3, train = 0.7))
biden_train70 = biden_split7030$train %>%
                tbl_df()
biden_test30 = biden_split7030$test %>%
               tbl_df()

biden_bag_data_train = biden_train70 %>%
                       select(-Party) %>%
```

```
                      mutate_each(funs(as.factor(.)), dem, rep) %>%
                      na.omit
```

```
## `mutate_each()` is deprecated.
## Use `mutate_all()`, `mutate_at()` or `mutate_if()` instead.
## To map `funs` over a selection of variables, use `mutate_at()`
```

```
biden_bag_data_test = biden_test30 %>%
                      select(-Party) %>%
                      mutate_each(funs(as.factor(.)), dem, rep) %>%
                      na.omit
```

```
## `mutate_each()` is deprecated.
## Use `mutate_all()`, `mutate_at()` or `mutate_if()` instead.
## To map `funs` over a selection of variables, use `mutate_at()`
```

```
# estimate model
(bag_biden <- randomForest(biden ~ ., data = biden_bag_data_train, mtry = 5, ntree = 500, importance=TRU
```

```
##
## Call:
##  randomForest(formula = biden ~ ., data = biden_bag_data_train,      mtry = 5, ntree = 500, importanc
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 5
##
##           Mean of squared residuals: 497
##                     % Var explained: 8.91
```

```
# find MSE
mse_bag_biden = mse(bag_biden, biden_bag_data_test)
```

The MSE for this model is 406, significantly higher than the previous, simpler model. The model only explains 8.91% of the variation which is low.

D).

```
set.seed(1234)

(biden_rf1 <- randomForest(biden ~ ., data = biden_bag_data_train,mtry =1,ntree = 500))
```

```
##
## Call:
##  randomForest(formula = biden ~ ., data = biden_bag_data_train,      mtry = 1, ntree = 500)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 1
##
##           Mean of squared residuals: 404
##                     % Var explained: 26
```

```
mse_rf = mse(biden_rf1, biden_bag_data_test)
(biden_rf2 <- randomForest(biden ~ ., data = biden_bag_data_train,mtry =2,ntree = 500))
```

```
##
## Call:
```

```
##  randomForest(formula = biden ~ ., data = biden_bag_data_train,      mtry = 2, ntree = 500)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##           Mean of squared residuals: 403
##                     % Var explained: 26.2
```

```
mse_rf = mse(biden_rf2, biden_bag_data_test)
(biden_rf3 <- randomForest(biden ~ ., data = biden_bag_data_train,mtry =3,ntree = 500))
```

```
##
## Call:
##  randomForest(formula = biden ~ ., data = biden_bag_data_train,      mtry = 3, ntree = 500)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##           Mean of squared residuals: 439
##                     % Var explained: 19.7
```

```
mse_rf = mse(biden_rf3, biden_bag_data_test)

rf_biden_importance = as.data.frame(importance(biden_rf1))
rf_biden_importance = as.data.frame(importance(biden_rf2))
rf_biden_importance = as.data.frame(importance(biden_rf3))
```

The MSEs are 404, 403, and 439. The first two MSEs are very similar (the former is slightly better than the latter) and the final one is obviously larger than both of them. This is because we only have 5 features so that the optimal parameter for random forest should be less than 2. In other words, other good features' importances are suppressed.

When m = 1 or 2, party is very important, and age, educ and female are unimportant. However, when m = 3, the results of importance are very similar to those in the bagging model: the importance of age is much larger than dem and GOP, and educcation is also moderately important. These results demonstrate age (and sometimes probabily education) is not a good predictor but has enough importance in a single tree. In this case, when we set a small m, a large percent of trees in random forest will not be based on important but bad features and thus has a better prediction.