Today

- Ensemble methods
  - Bagging
  - Random forest

Random forest

Algorithm

for many repetitions

    randomly select m predictor variables

    resample the data (rows) with replacement

    train the tree model

    record prediction

final prediction = mean of predictions

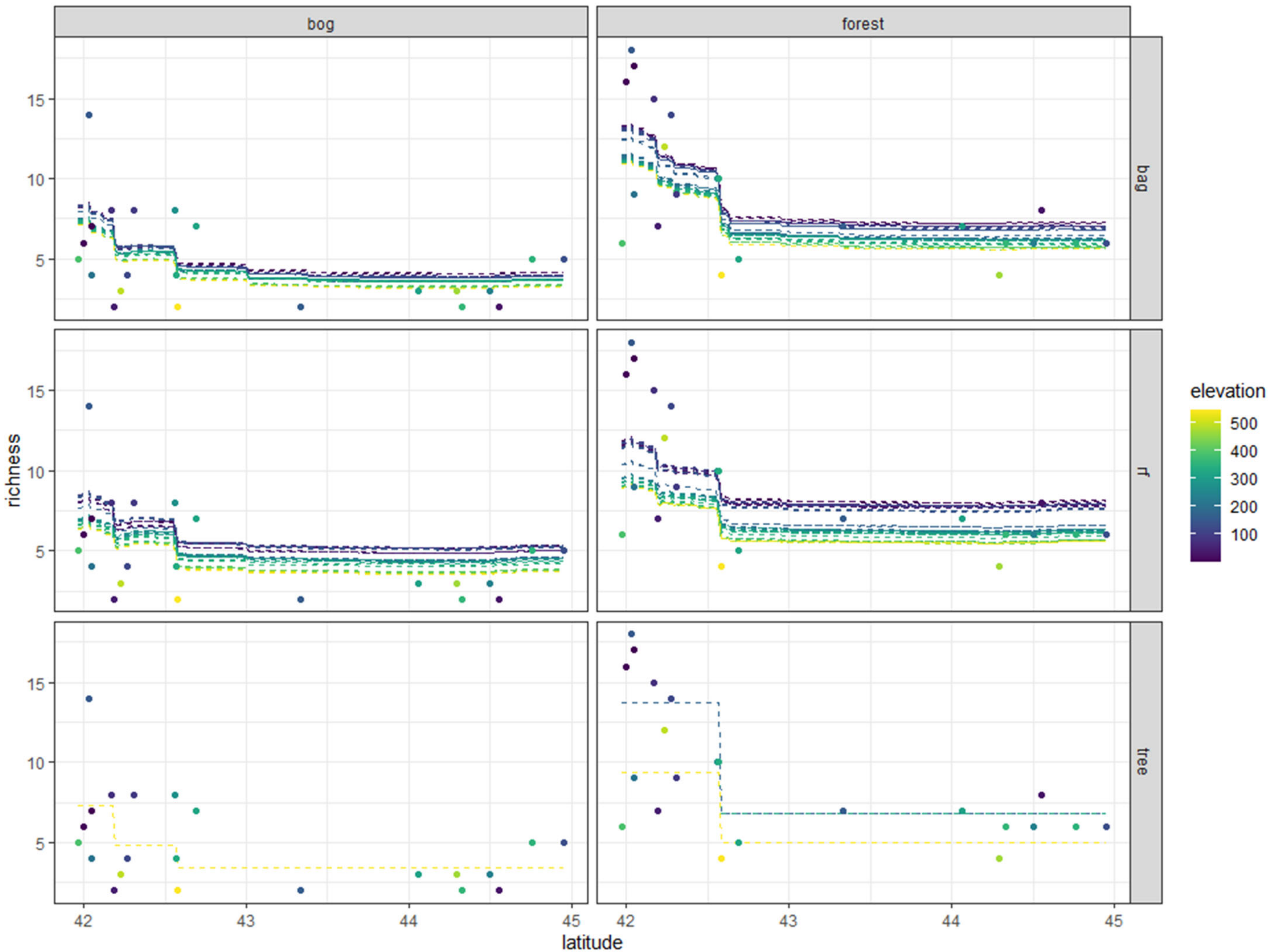
Random forest as R

```r
# Parameters
m <- 2 #Number of predictors to sample at each iteration
boot_reps <- 500

# Setup
n <- nrow(ants)
c <- ncol(ants)
nn <- nrow(grid_data)
boot_preds <- matrix(rep(NA, nn*boot_reps), nrow=nn, ncol=boot_reps)

# Main algorithm
for ( i in 1:boot_reps ) {
#   randomly select m predictor variables
    predictor_indices <- sample(2:c, m)
    boot_data <- ants[,c(1,predictor_indices)]
#   resample the data (rows) with replacement
    boot_indices <- sample(1:n, n, replace=TRUE)
    boot_data <- boot_data[boot_indices,]
#   train the tree model
    boot_fit <- tree(richness ~ ., data=boot_data)
#   record prediction
    boot_preds[,i] <- predict(boot_fit, newdata=grid_data)
}
rf_preds <- rowMeans(boot_preds)
```

the new bit


bog
forest
bag
rf
tree
richness
latitude
elevation
500
400
300
200
100
15
10
5
42
43
44
45

Inference algorithm

- k-fold CV
  - expensive, as we’ve seen
  - use for comparison with other models
- Out-of-bag estimate
  - can use for tuning

“Out of bag” error estimate

- Each bootstrap we can use the other samples to gauge prediction error
- Approx equal to LOOCV
- Computationally efficient
  - trivial to add to the bagging step

Variable importance

- Average RSS decrease over splits for each variable
- Interpretation:
  - more reduction in RSS = more “important”
- Similar to regression concept of “explaining more variation”
- Advantage: explainable machine learning