An Introduction to Regression Trees (CART)

Susan Athey, Stanford University

Machine Learning and Causal Inference

What is the goal of prediction?

- Machine learning answer:
  - Smallest mean-squared error in a test set

- Formally:
  - Let S^{te} be a test set.
    - Think of this as a random draw of individuals from a population
  - Let $\hat{\mu}(x_i)$ be a candidate (estimated) predictor
  - MSE on test set is:

$$\frac{1}{|S^{te}|}\sum_{i \in S^{te}} (Y_i - \hat{\mu}(X_i))^2$$

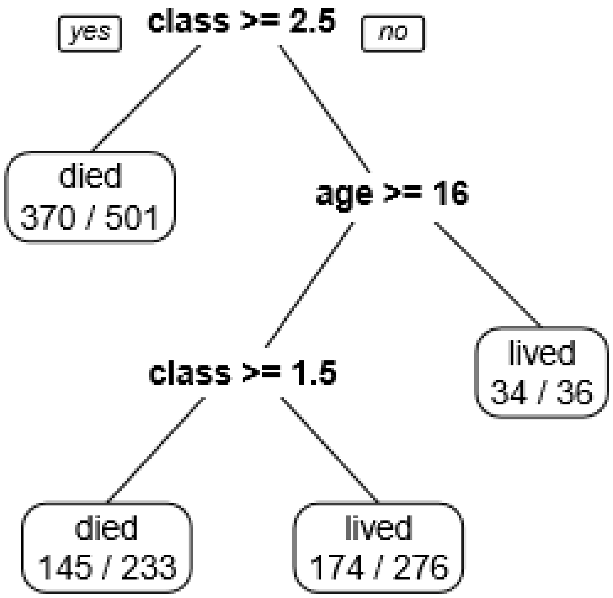
Regression Trees

- Simple method for prediction
  - Partition data into subsets by covariates
  - Predict using average within each subset

- Why are regression trees popular?
  - Easy to understand and explain
  - Businesses often need "segments"
  - Software assigns different algorithms to different segments

- Can completely describe the algorithm and interpretation

Example: Who survived the Titantic?



Regression Trees for Prediction

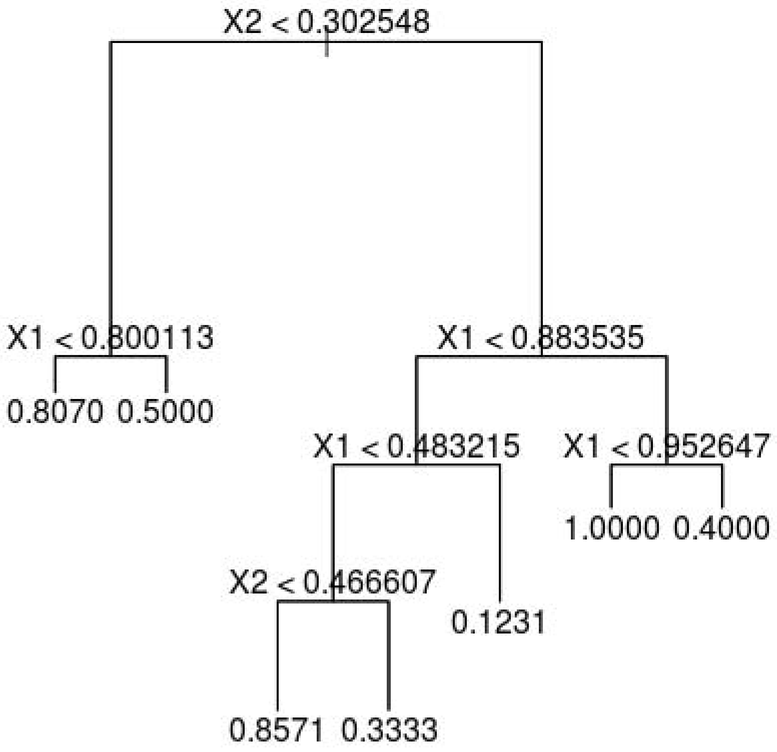
Data

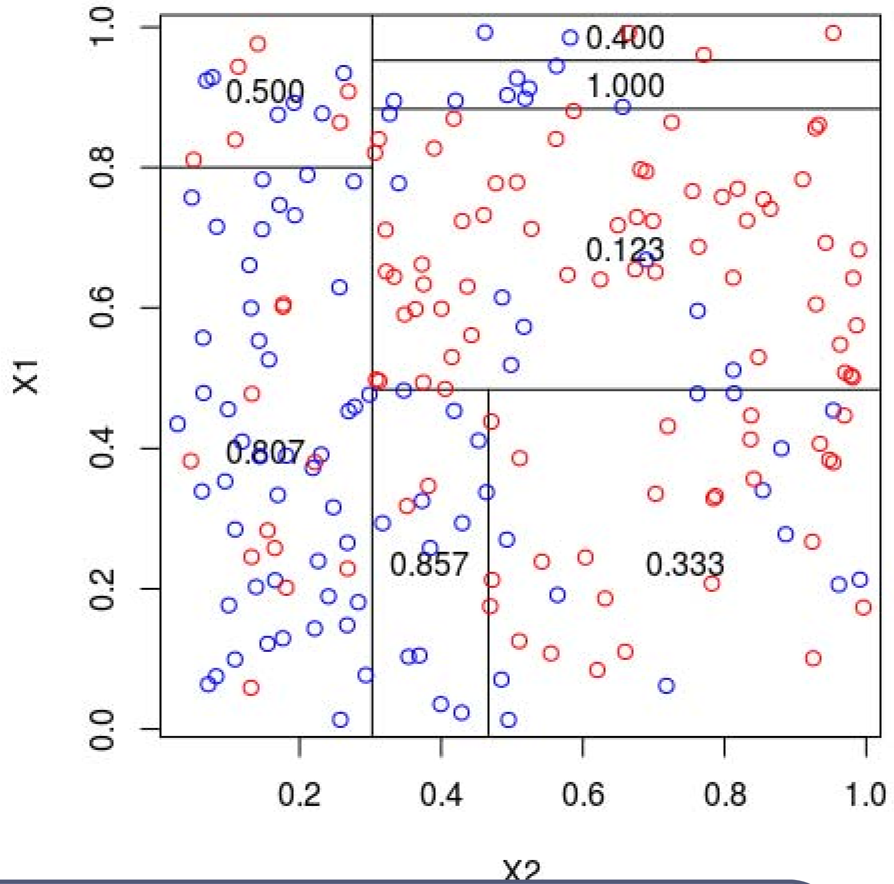
- Outcomes Y_i, attributes X_i.
- Support of X_i is $\mathcal{X}$.
- Have training sample with independent obs.
- Want to predict on new sample

Build a "tree":

- Partition of $\mathcal{X}$ into "leaves" $\mathcal{X}_j$
- Predict Y conditional on realization of X in each region $\mathcal{X}_j$ using the sample mean in that region
- Go through variables and leaves and decide whether and where to split leaves (creating a finer partition) using in-sample goodness of fit criterion
- Select tree complexity using cross-validation based on prediction quality

Regression Trees for Prediction





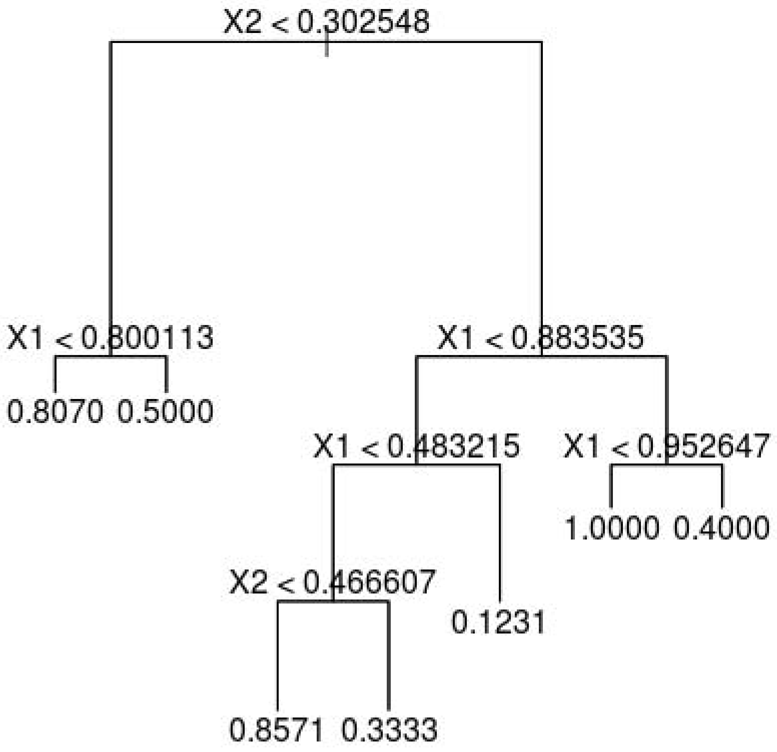
Outcome: Binary (Y in {0,1})

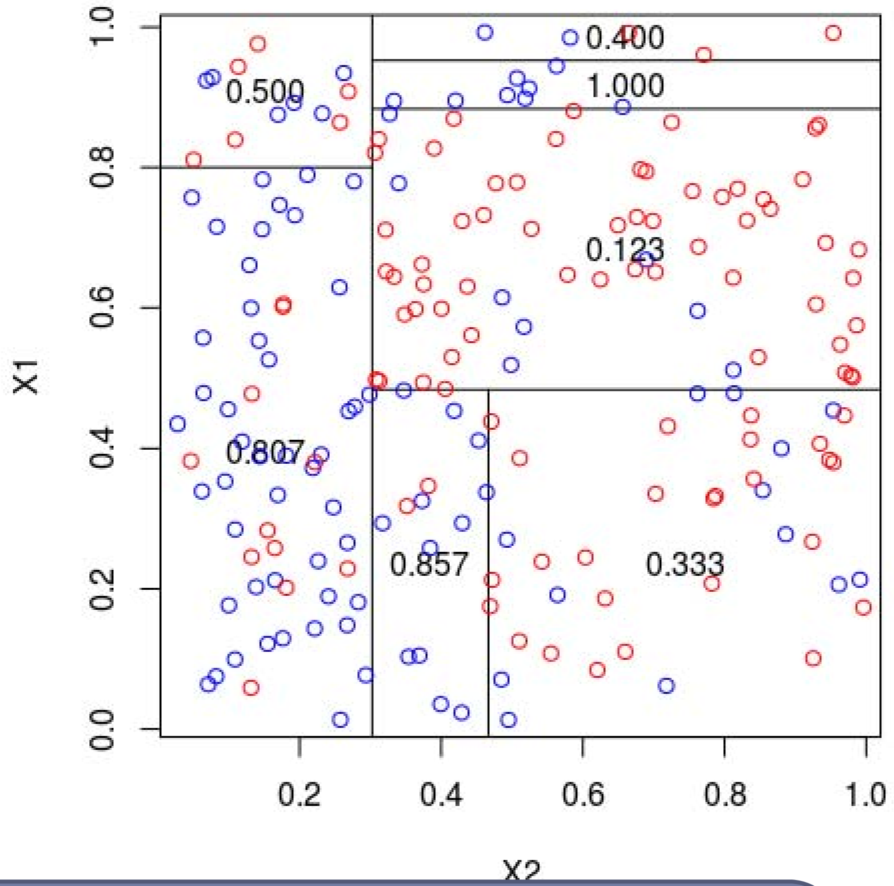
Two covariates

Goal: Predict Y as a function of X

"Classify" units as a function of X according to whether they are more likely to have Y=0 or Y=1

Regression Trees for Prediction





(I) Tree-building: Use algorithm to partition data according to covariates (adaptive: do this based on the difference in mean outcomes in different potential leaves.)

(II) Estimation/prediction: calculate mean outcomes in each leaf

(III) Use cross-validation to select tree complexity penalty

Tree Building Details

- Impossible to search over all possible partitions, so use a greedy algorithm
- Do until all leaves have less than 2*minsize obs:
  - For each leaf:
    - For each observed value $\breve{x}_j$ of each covariate x_j:
      - Consider splitting the leaf into two children according to whether $\breve{x}_j \leq x_j$
      - Make new predictions in each candidate child according to sample mean
      - Calculate the improvement in "fit" (MSE)
    - Select the covariate *j* and the cutoff value that lead to the greatest improvement in MSE; split the leaf into two child leaves
- Observations
  - In-sample MSE always improves with additional splits
  - What is MSE when each leaf has one observation?

Problem: Tree has been “over-fitted”

- Suppose we fit a tree and pick a particular leaf ℓ.
  - Do we expect that if we drew a new sample, we would get the same answer?

- More formally:
  - Let S^{tr} be training dataset and S^{te} be an independent test set
  - Let $\widehat{\mu}(x_i) = \frac{1}{N_{\ell(x_i)}} \sum_{i \in \ell(x_i), S^{tr}} Y_i$
  - Is $E_{i \in S^{te}}[Y_i | X_i \in \ell(x_i)] = \hat{\mu}(x_i)$?

What are tradeoffs in tree depth?

- First: note that in-sample MSE doesn't guide you
  - It always increases with depth

- Tradeoff as you grow tree deeper
  - More personalized predictions
  - More biased estimates

Regression Trees for Prediction: Components

1. Model and Estimation
   - A. Model type: Tree structure
   - B. Estimator $\hat{Y}_i$: sample mean of Y_i within leaf
   - C. Set of candidate estimators *C*: correspond to different specifications of how tree is split

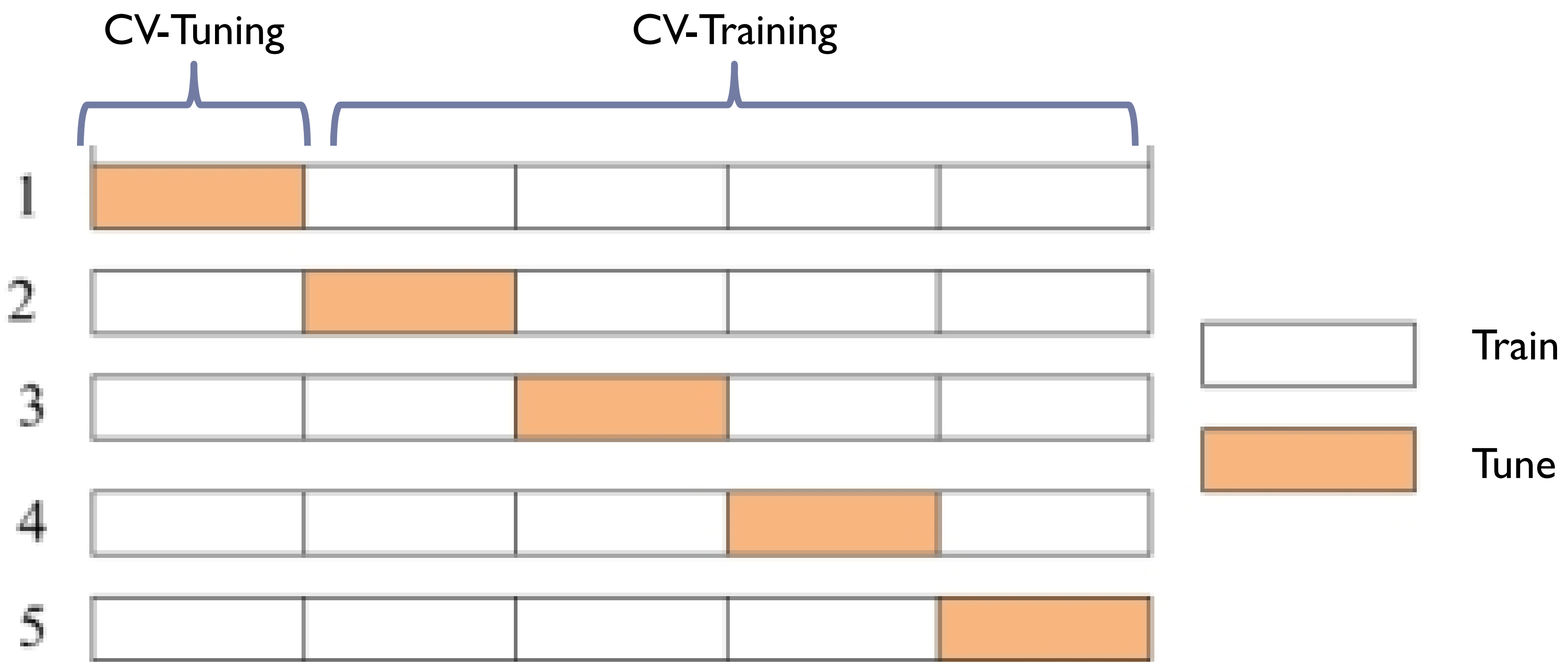
2. Criterion function (for fixed tuning parameter λ)
   - A. In-sample Goodness-of-fit function:

     $$Q^{is} = \text{-MSE (Mean Squared Error)} = -\frac{1}{N}\sum_{i=1}^{N}\left(\hat{Y}_i - Y_i\right)^2$$

   - A. Structure and use of criterion
     - i. Criterion: $Q^{crit} = Q^{is} - \lambda \times \#$ leaves
     - ii. Select member of set of candidate estimators that maximizes Q^{crit}, given λ

3. Cross-validation approach
   - A. Approach: Cross-validation on grid of tuning parameters. Select tuning parameter λ with highest Out-of-sample Goodness-of-Fit Q^{os}.
   - B. Out-of-sample Goodness-of-fit function: $Q^{os} = \text{-MSE}$

How Does Cross Validation Work?



Tuning Set = 1/5 of Training Set

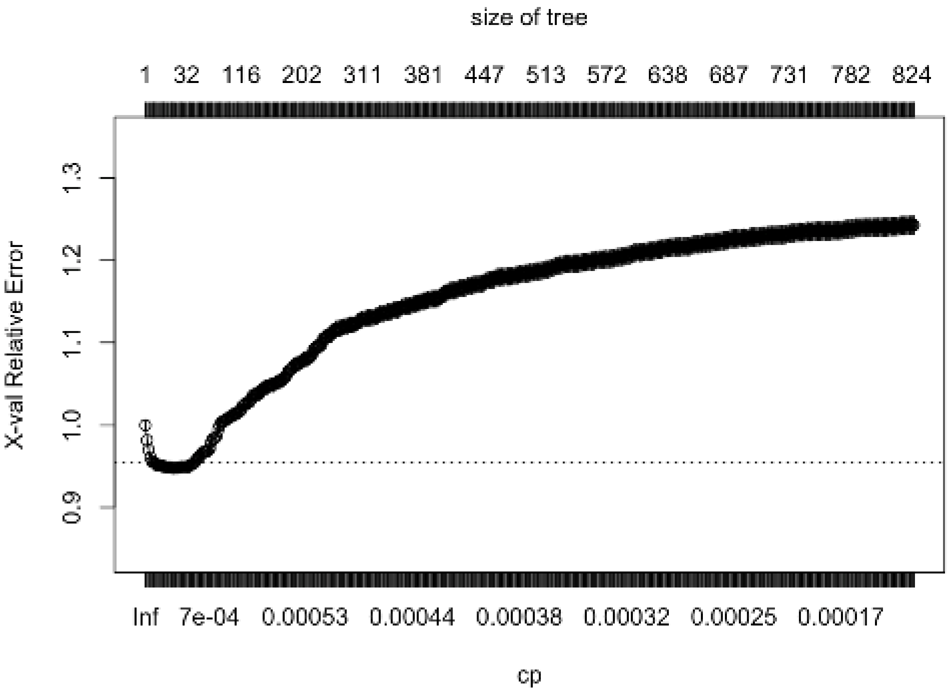
Cross-Validation Mechanics

- Loop over cross-validation samples
  - Train a deep tree on CV-training subset

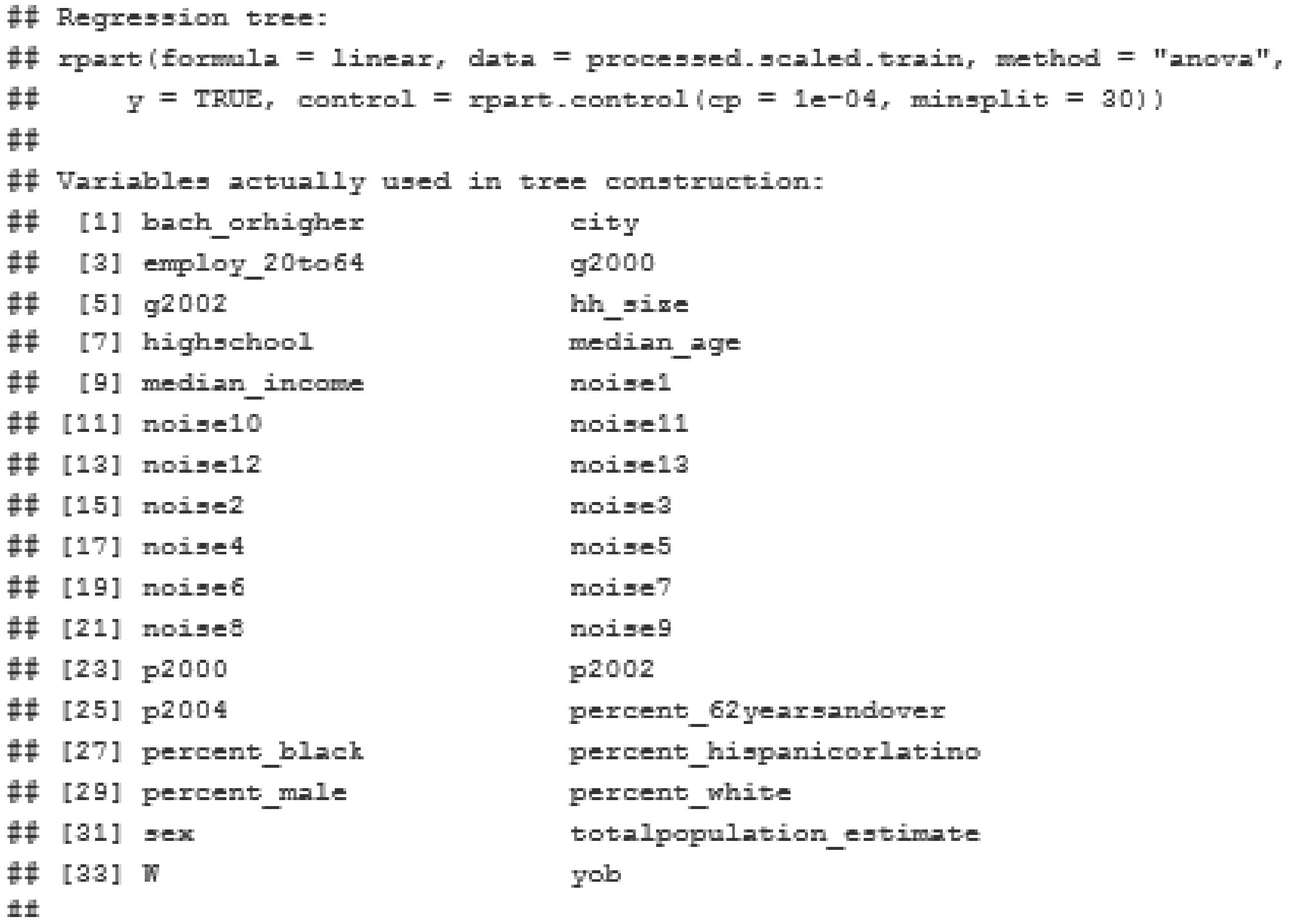
- Loop over penalty parameters λ
  - Loop over cross-validation samples
    - Prune the tree according to penalty
    - Calculate new MSE of tree
  - Average (over c-v samples) the MSE for this penalty

- Choose the penalty λ^* that gives the best average MSE

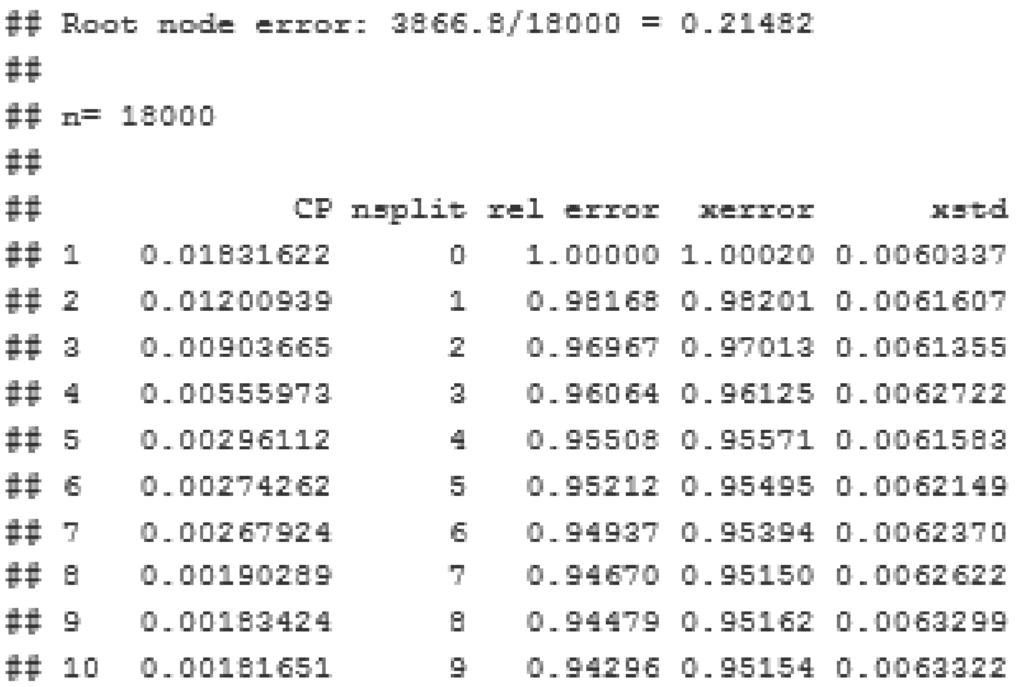
Choosing the penalty parameter



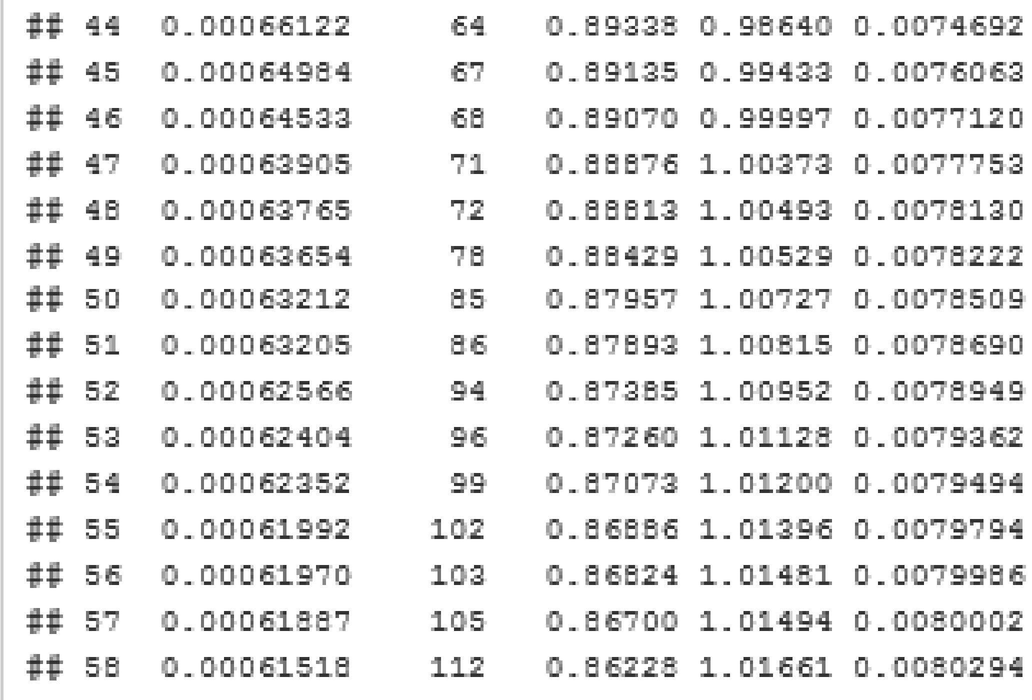
Some example code



```
## Regression tree:
## rpart(formula = linear, data = processed.scaled.train, method = "anova",
##     y = TRUE, control = rpart.control(cp = 1e-04, minsplit = 30))
##
## Variables actually used in tree construction:
##  [1] bach_orhigher            city
##  [3] employ_20to64            g2000
##  [5] g2002                    hh_size
##  [7] highschool               median_age
##  [9] median_income            noise1
## [11] noise10                  noise11
## [13] noise12                  noise13
## [15] noise2                   noise3
## [17] noise4                   noise5
## [19] noise6                   noise7
## [21] noise8                   noise9
## [23] p2000                    p2002
## [25] p2004                    percent_62yearsandover
## [27] percent_black            percent_hispanicorlatino
## [29] percent_male             percent_white
## [31] sex                      totalpopulation_estimate
## [33] W                        yob
##
```

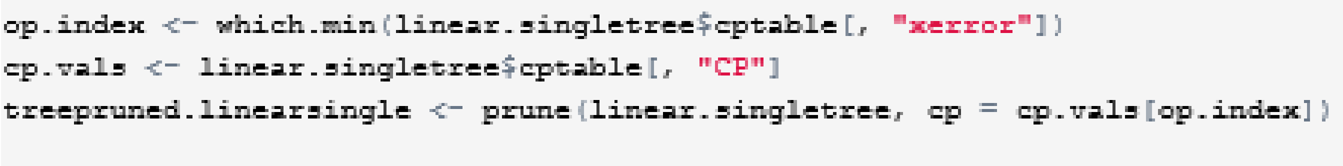


```
## Root node error: 3866.8/18000 = 0.21482
##
## n= 18000
##
##              CP nsplit rel error  xerror      xstd
## 1   0.01831622      0   1.00000 1.00020 0.0060337
## 2   0.01200939      1   0.98168 0.98201 0.0061607
## 3   0.00903665      2   0.96967 0.97013 0.0061355
## 4   0.00555973      3   0.96064 0.96125 0.0062722
## 5   0.00296112      4   0.95508 0.95571 0.0061583
## 6   0.00274262      5   0.95212 0.95495 0.0062149
## 7   0.00267924      6   0.94937 0.95394 0.0062370
## 8   0.00190289      7   0.94670 0.95150 0.0062622
## 9   0.00183424      8   0.94479 0.95162 0.0063299
## 10  0.00181651      9   0.94296 0.95154 0.0063322
```



```
## 44  0.00066122     64   0.89338 0.98640 0.0074692
## 45  0.00064984     67   0.89135 0.99433 0.0076063
## 46  0.00064533     68   0.89070 0.99997 0.0077120
## 47  0.00063905     71   0.88876 1.00373 0.0077753
## 48  0.00063765     72   0.88813 1.00493 0.0078130
## 49  0.00063654     78   0.88429 1.00529 0.0078222
## 50  0.00063212     85   0.87957 1.00727 0.0078509
## 51  0.00063205     86   0.87893 1.00815 0.0078690
## 52  0.00062566     94   0.87385 1.00952 0.0078949
## 53  0.00062404     96   0.87260 1.01128 0.0079362
## 54  0.00062352     99   0.87073 1.01200 0.0079494
## 55  0.00061992    102   0.86886 1.01396 0.0079794
## 56  0.00061970    103   0.86824 1.01481 0.0079986
## 57  0.00061887    105   0.86700 1.01494 0.0080002
## 58  0.00061518    112   0.86228 1.01661 0.0080294
```

Pruning Code



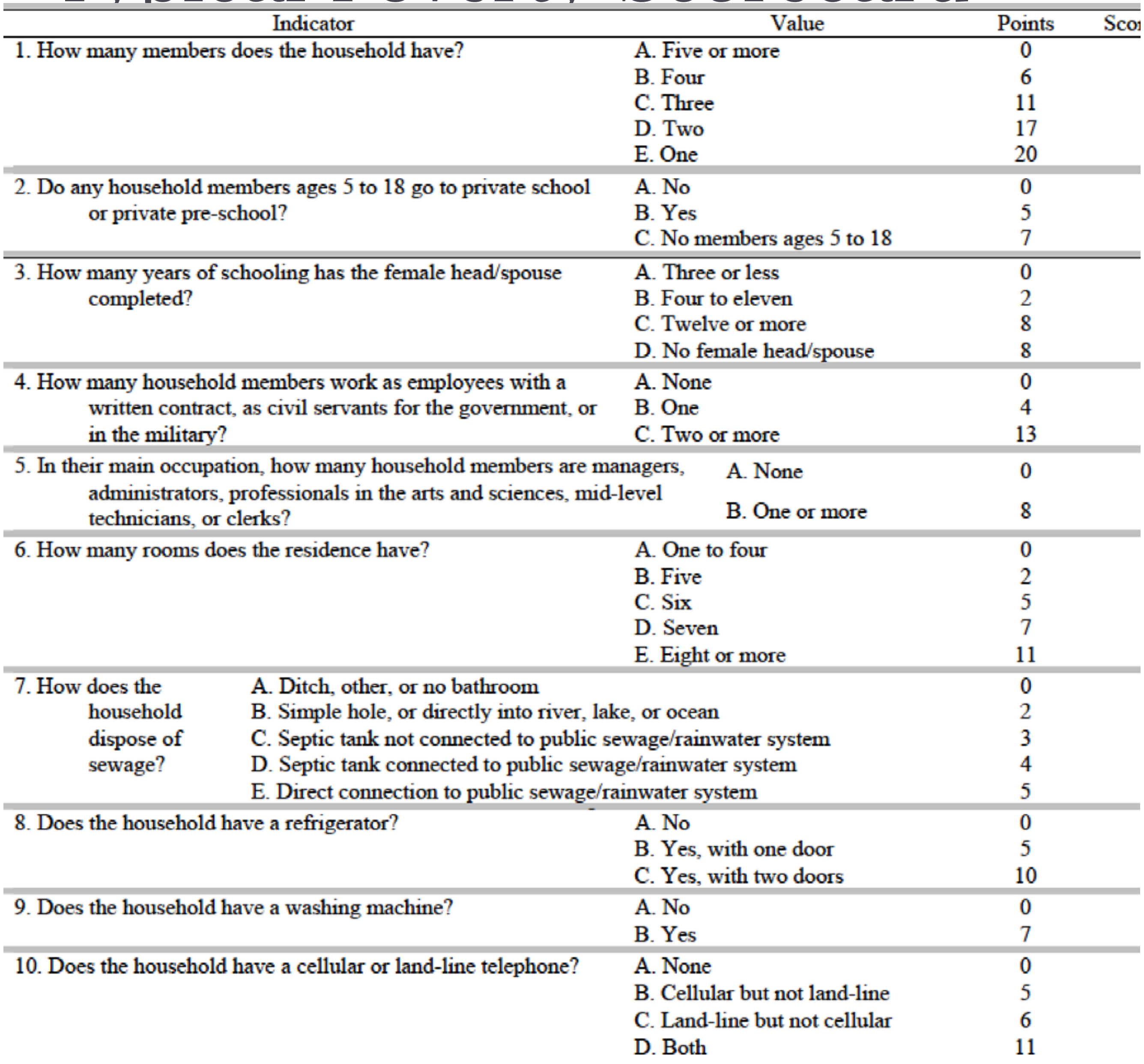
```
op.index <- which.min(linear.singletree$cptable[, "xerror"])
cp.vals <- linear.singletree$cptable[, "CP"]
treepruned.linearsingle <- prune(linear.singletree, cp = cp.vals[op.index])
```

A Basic Policy Problem

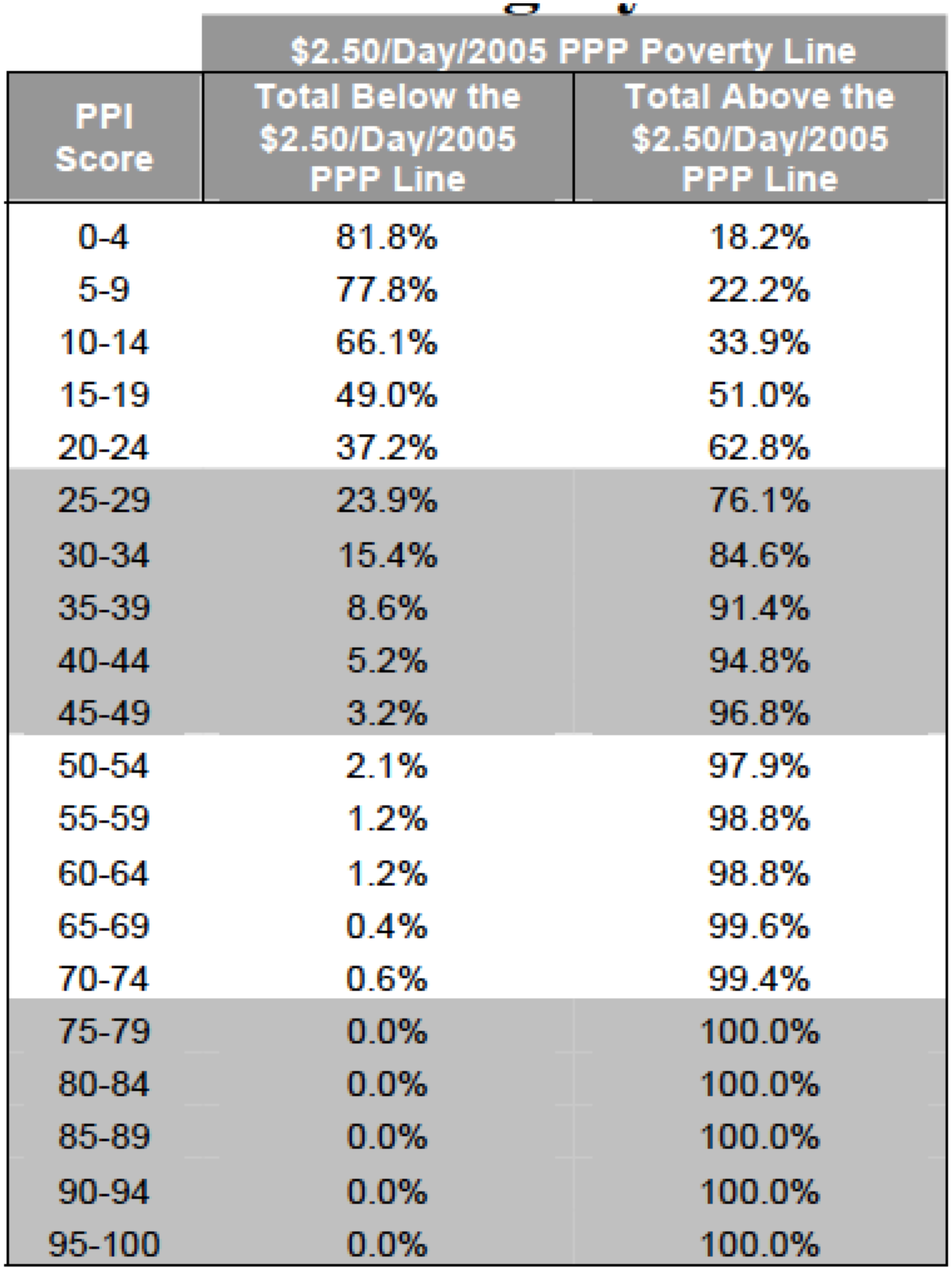
- Every transfer program in the world must determine…
  - Who is eligible for the transfer
- Typical goal of redistributive programs
  - Transfer to neediest
- But identifying the neediest is easier said than done

Thanks to Sendhil Mullainathan for providing this worked out example….

Typical Poverty Scorecard



Indicator	Value	Points	Sco
1. How many members does the household have?	A. Five or more	0	
	B. Four	6	
	C. Three	11	
	D. Two	17	
	E. One	20	
2. Do any household members ages 5 to 18 go to private school or private pre-school?	A. No	0	
	B. Yes	5	
	C. No members ages 5 to 18	7	
3. How many years of schooling has the female head/spouse completed?	A. Three or less	0	
	B. Four to eleven	2	
	C. Twelve or more	8	
	D. No female head/spouse	8	
4. How many household members work as employees with a written contract, as civil servants for the government, or in the military?	A. None	0	
	B. One	4	
	C. Two or more	13	
5. In their main occupation, how many household members are managers, administrators, professionals in the arts and sciences, mid-level technicians, or clerks?	A. None	0	
	B. One or more	8	
6. How many rooms does the residence have?	A. One to four	0	
	B. Five	2	
	C. Six	5	
	D. Seven	7	
	E. Eight or more	11	
7. How does the household dispose of sewage?	A. Ditch, other, or no bathroom	0	
	B. Simple hole, or directly into river, lake, or ocean	2	
	C. Septic tank not connected to public sewage/rainwater system	3	
	D. Septic tank connected to public sewage/rainwater system	4	
	E. Direct connection to public sewage/rainwater system	5	
8. Does the household have a refrigerator?	A. No	0	
	B. Yes, with one door	5	
	C. Yes, with two doors	10	
9. Does the household have a washing machine?	A. No	0	
	B. Yes	7	
10. Does the household have a cellular or land-line telephone?	A. None	0	
	B. Cellular but not land-line	5	
	C. Land-line but not cellular	6	
	D. Both	11	



	$2.50/Day/2005 PPP Poverty Line	
PPI Score	Total Below the $2.50/Day/2005 PPP Line	Total Above the $2.50/Day/2005 PPP Line
0-4	81.8%	18.2%
5-9	77.8%	22.2%
10-14	66.1%	33.9%
15-19	49.0%	51.0%
20-24	37.2%	62.8%
25-29	23.9%	76.1%
30-34	15.4%	84.6%
35-39	8.6%	91.4%
40-44	5.2%	94.8%
45-49	3.2%	96.8%
50-54	2.1%	97.9%
55-59	1.2%	98.8%
60-64	1.2%	98.8%
65-69	0.4%	99.6%
70-74	0.6%	99.4%
75-79	0.0%	100.0%
80-84	0.0%	100.0%
85-89	0.0%	100.0%
90-94	0.0%	100.0%
95-100	0.0%	100.0%

Can we do better?

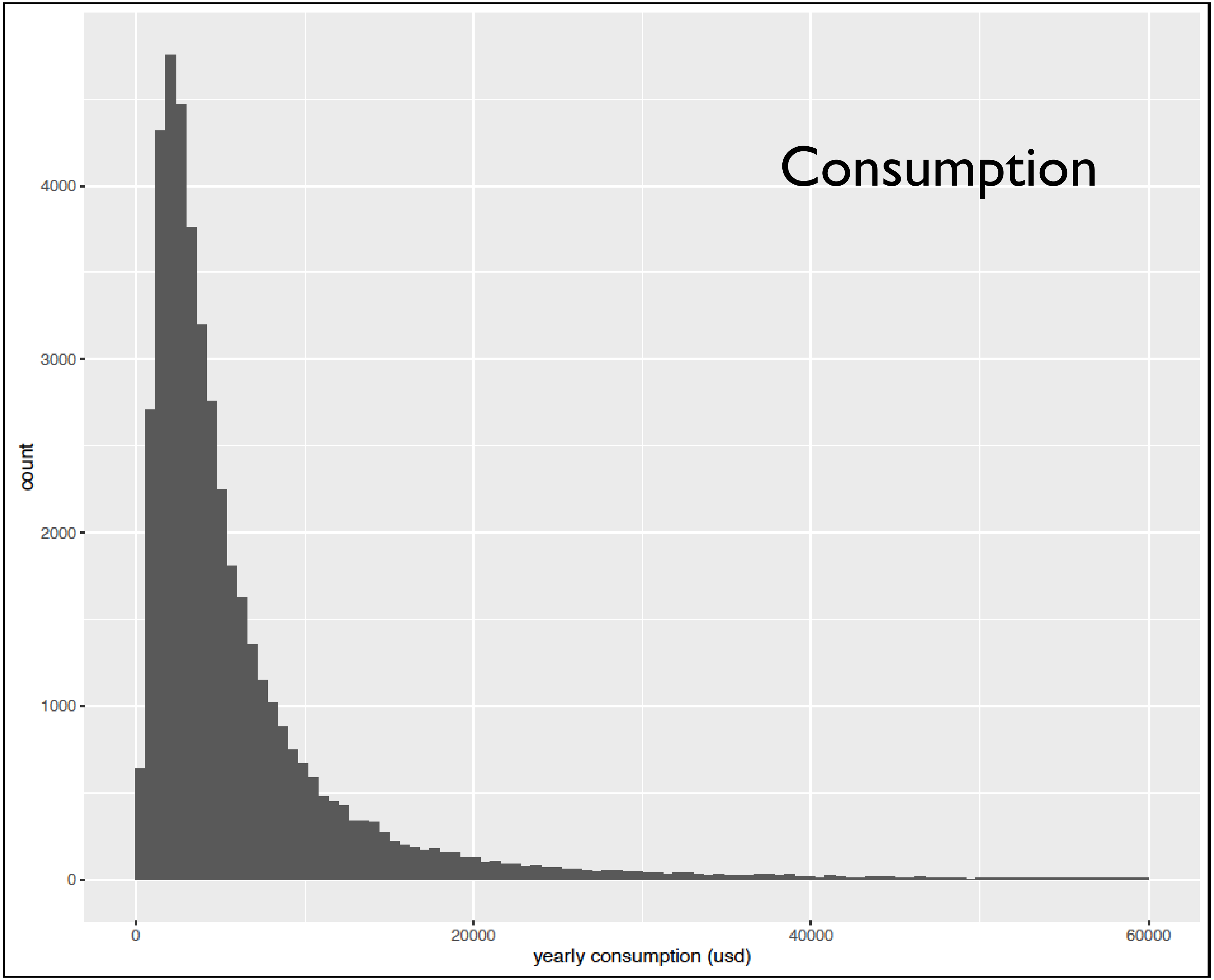
- This component of targeting is a pure prediction problem

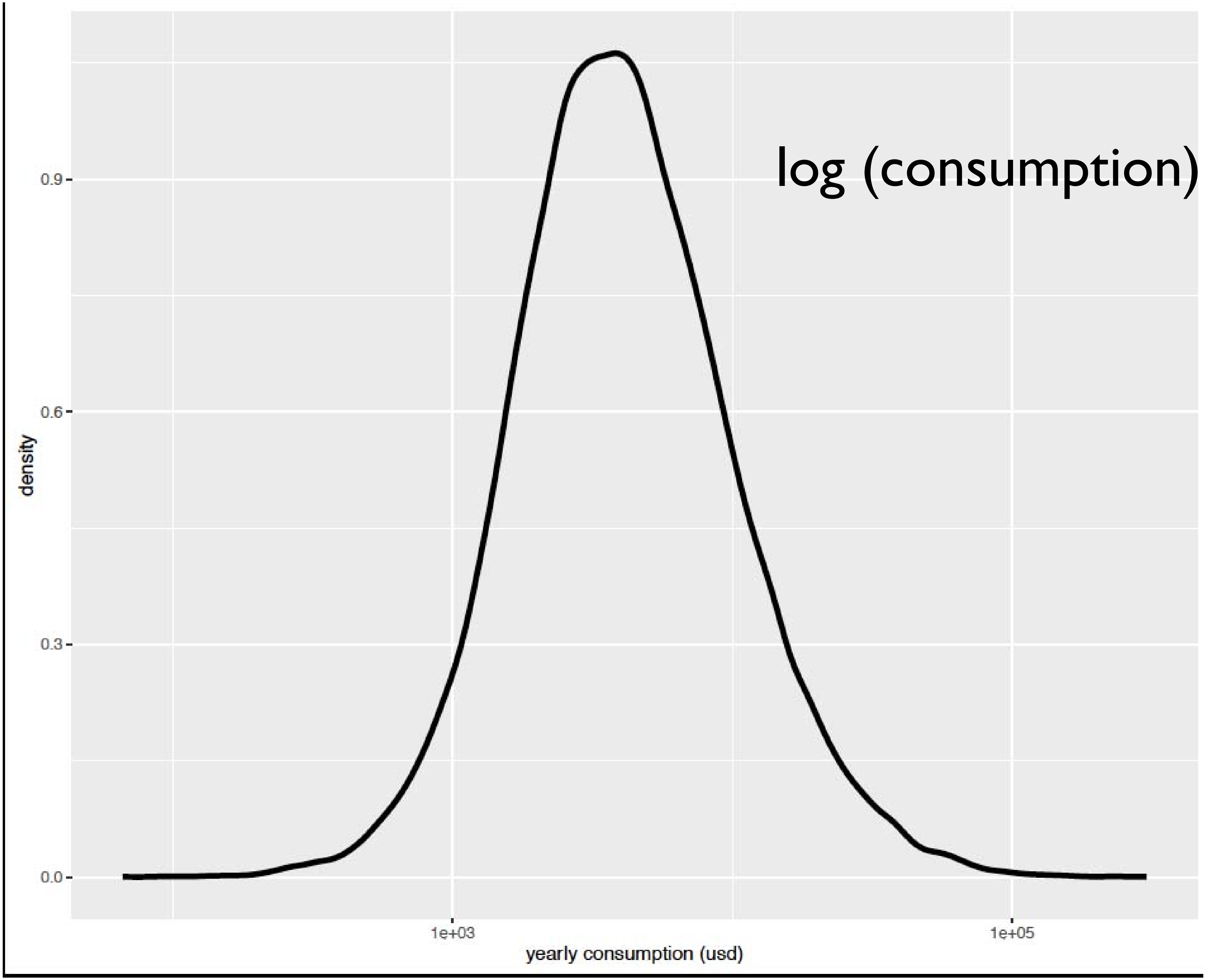
- We fundamentally care about getting best predictive accuracy

- Let’s use this example to illustrate the mechanics of prediction

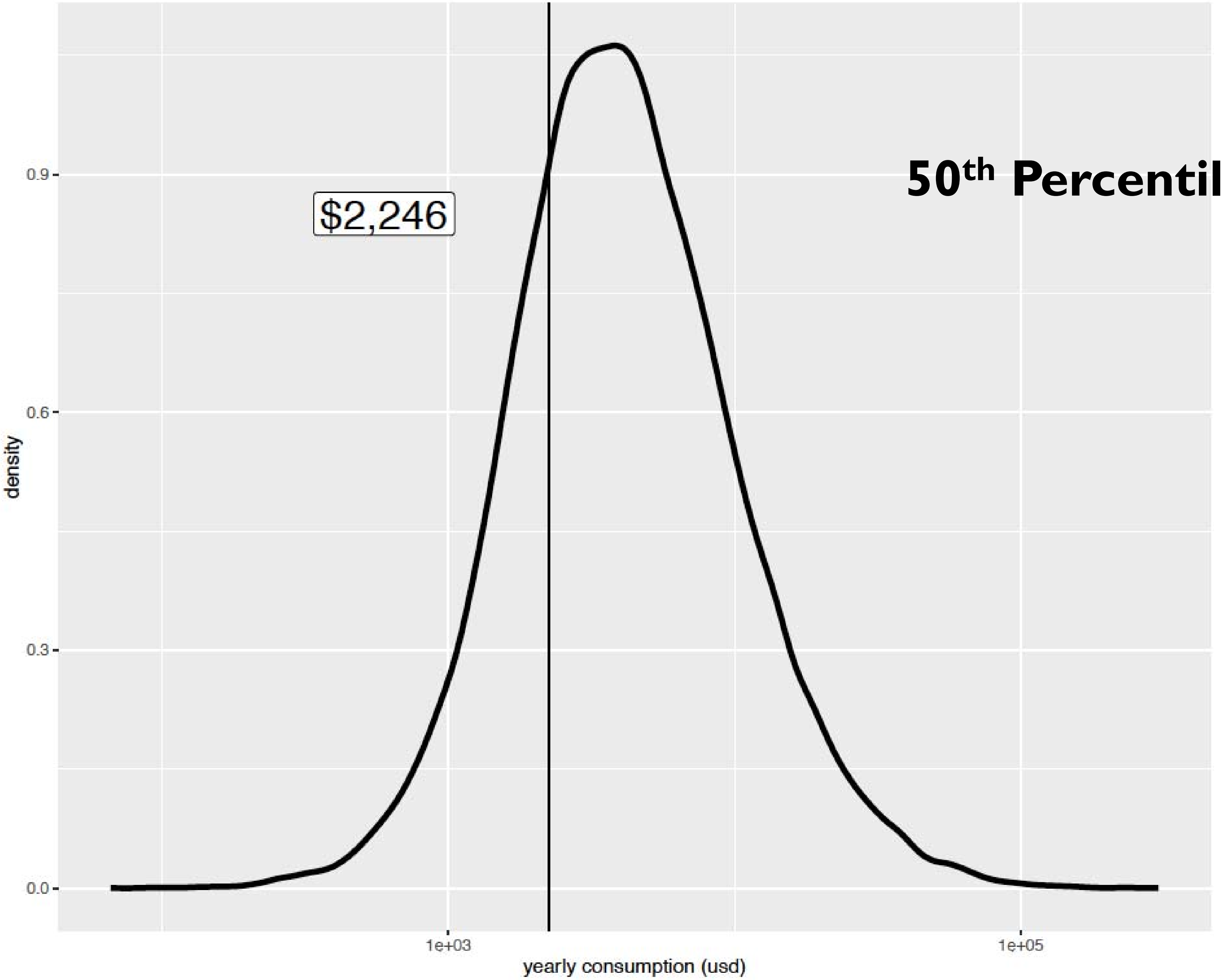
Brazilian Data

- The data:
  - 44,787 data points
  - 53 variables
  - Not very wide?
- Median
  - Annual consumption (in dollars): 3918
  - 348.85 monthly income
- 6 percent below 1.90 poverty line
- 14 percent below the 3.10 poverty line


Consumption
count
4000
3000
2000
1000
0
0
20000
40000
60000
yearly consumption (usd)

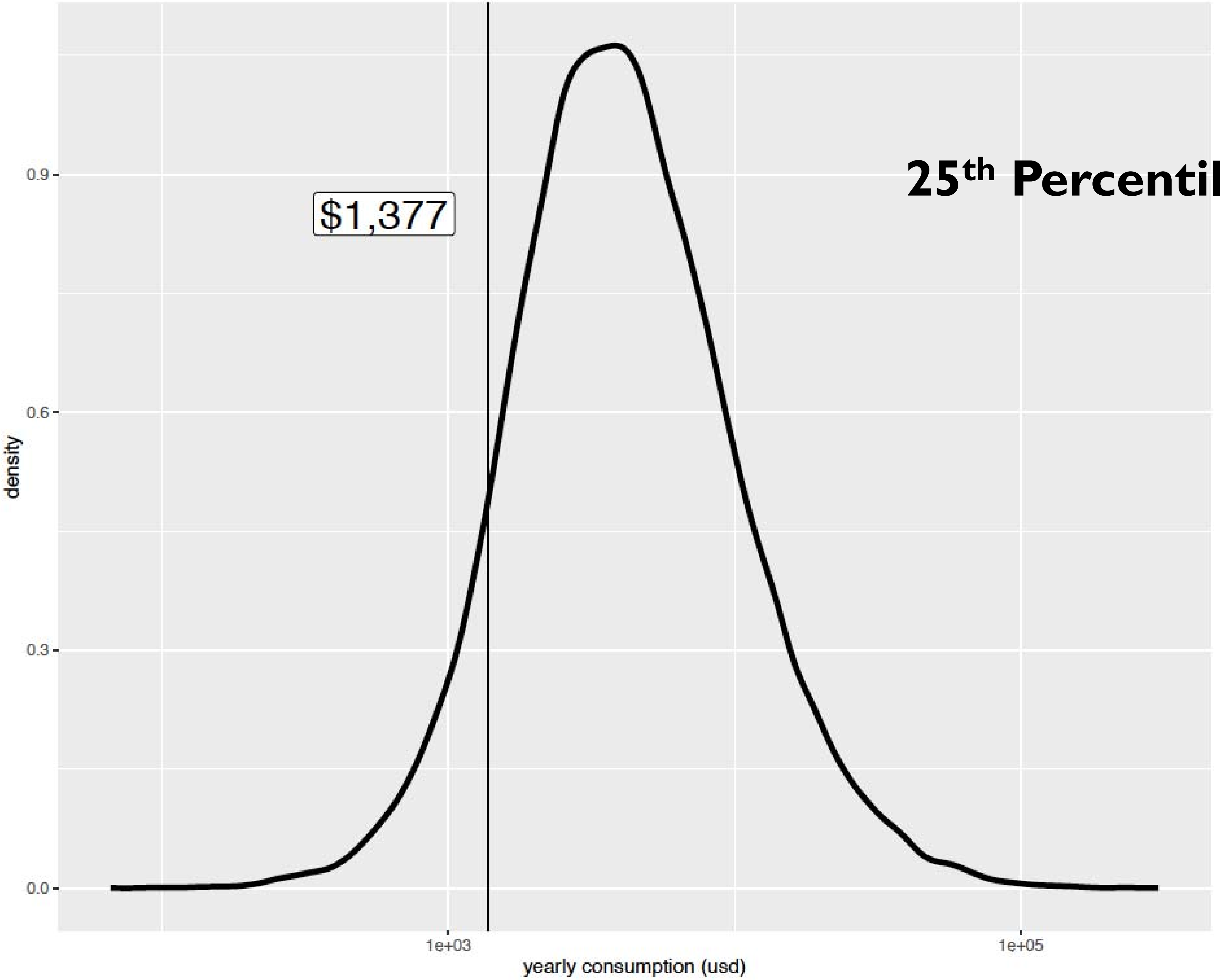


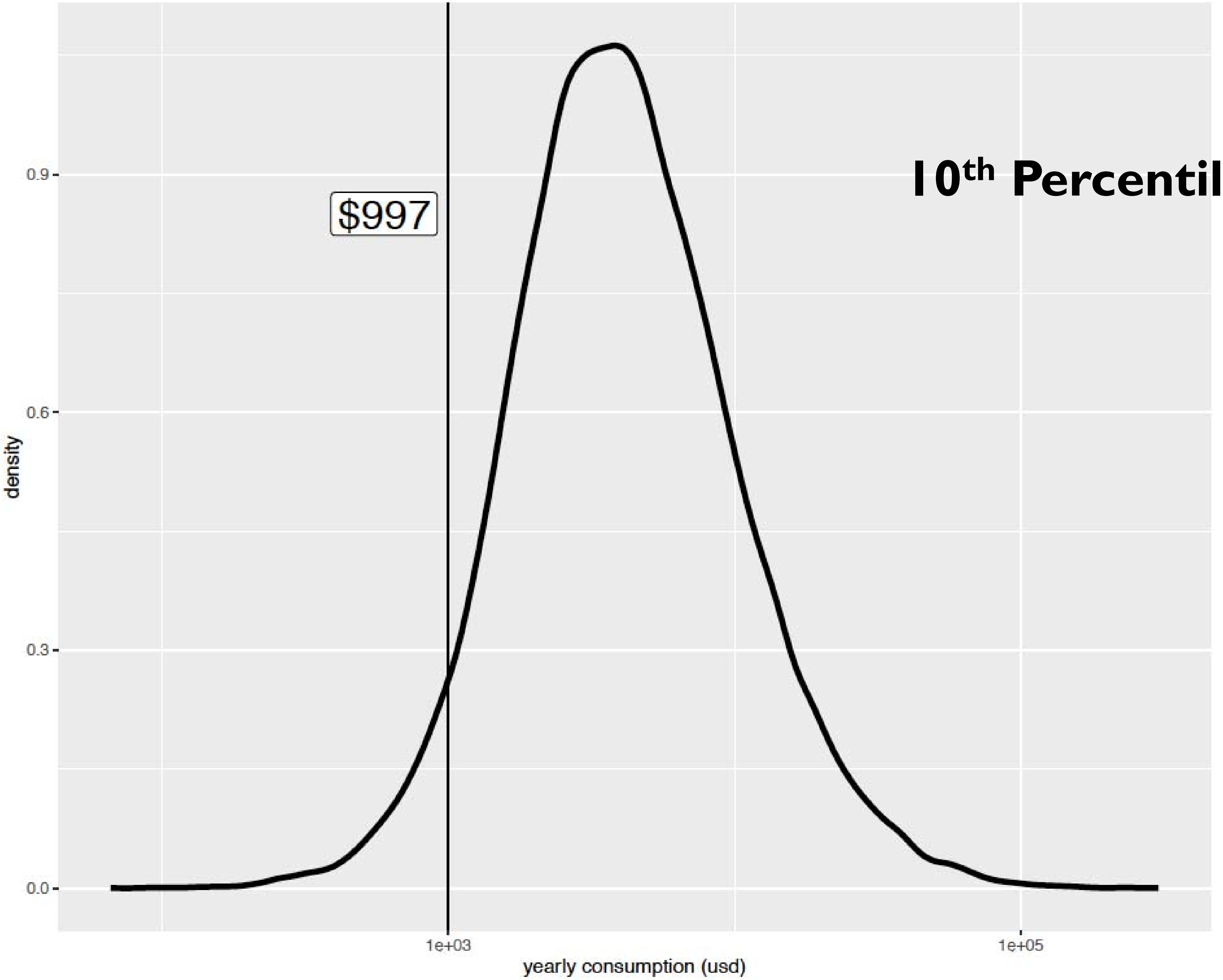
log (consumption)



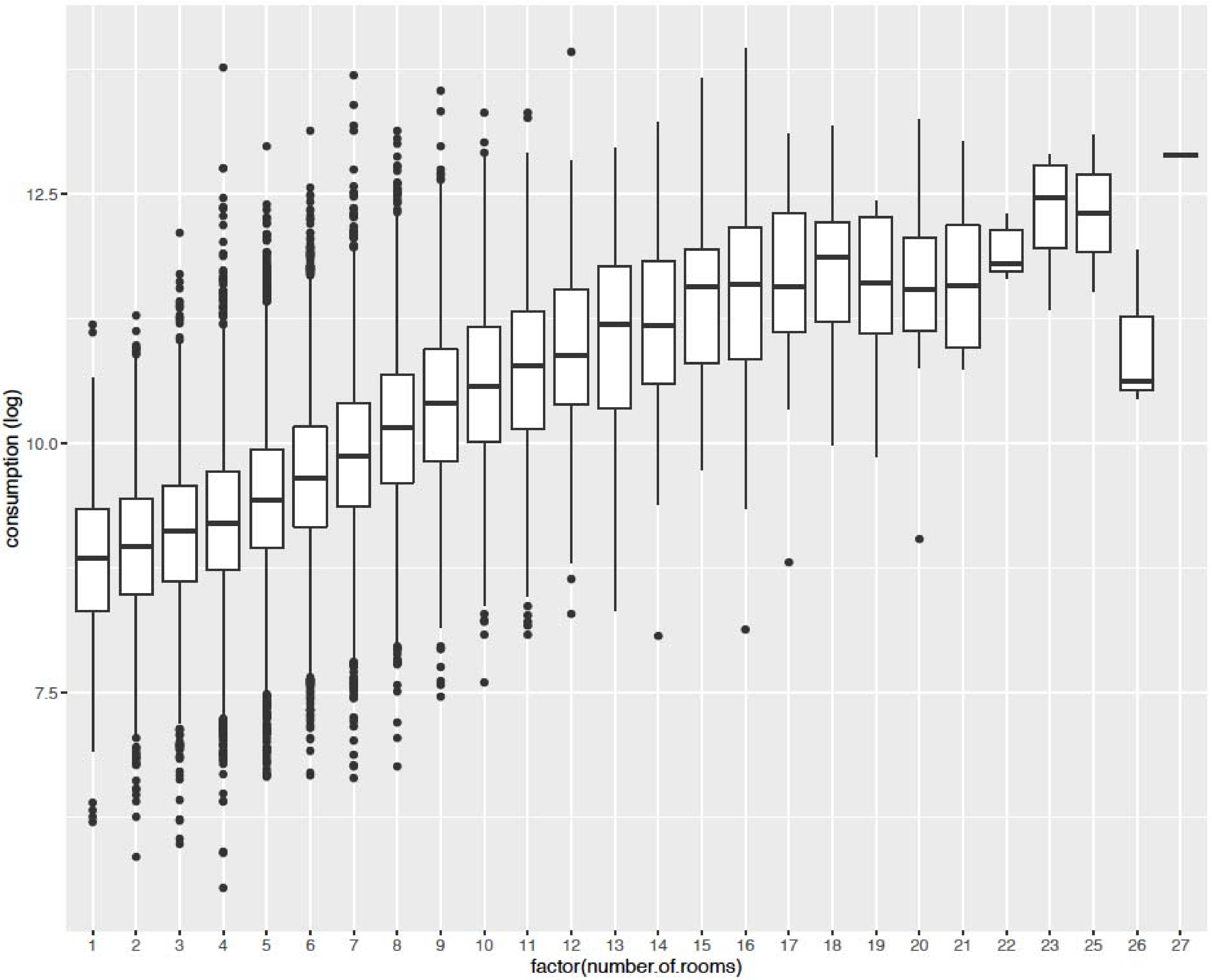
50th Percentile

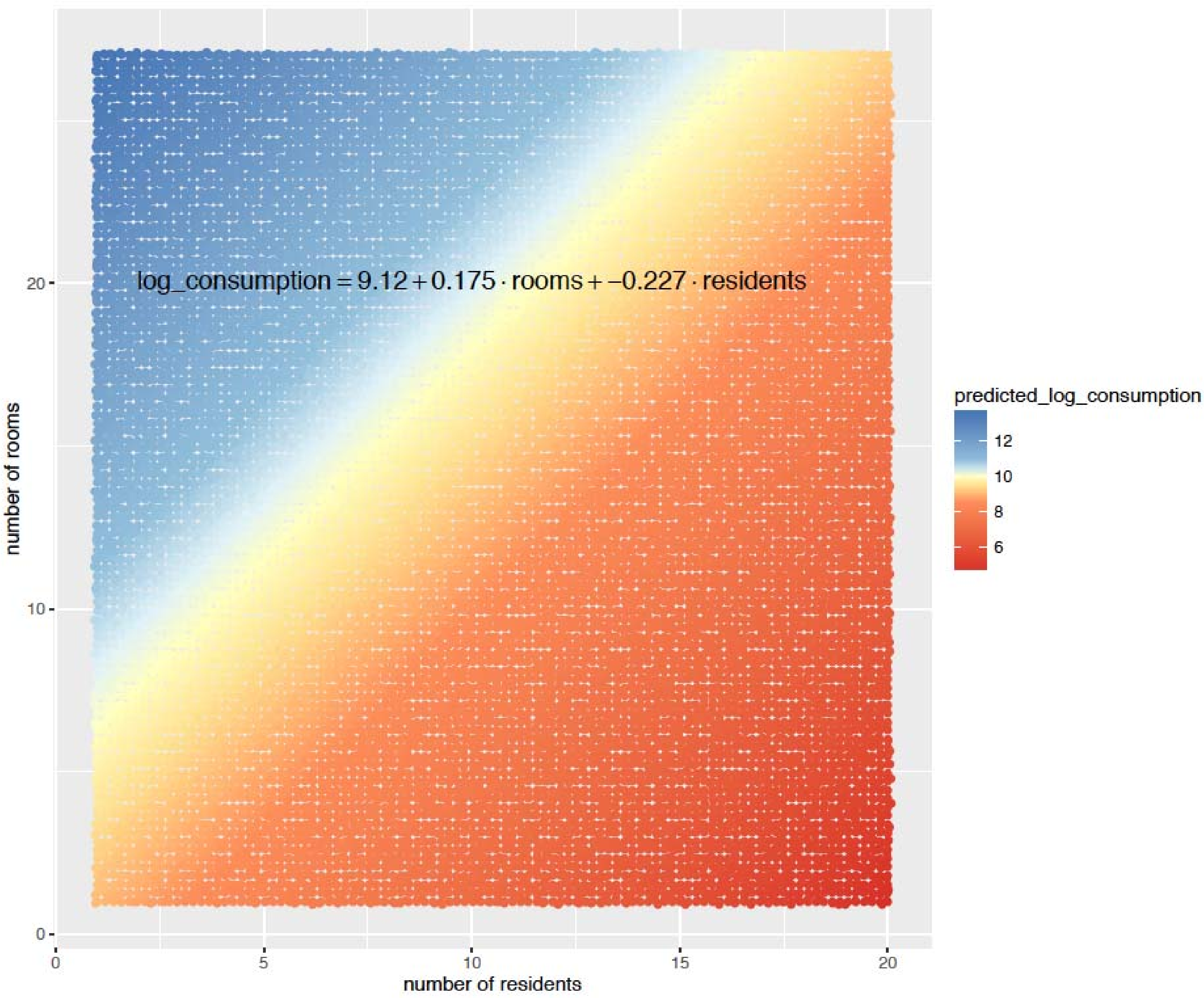
25th Percentile

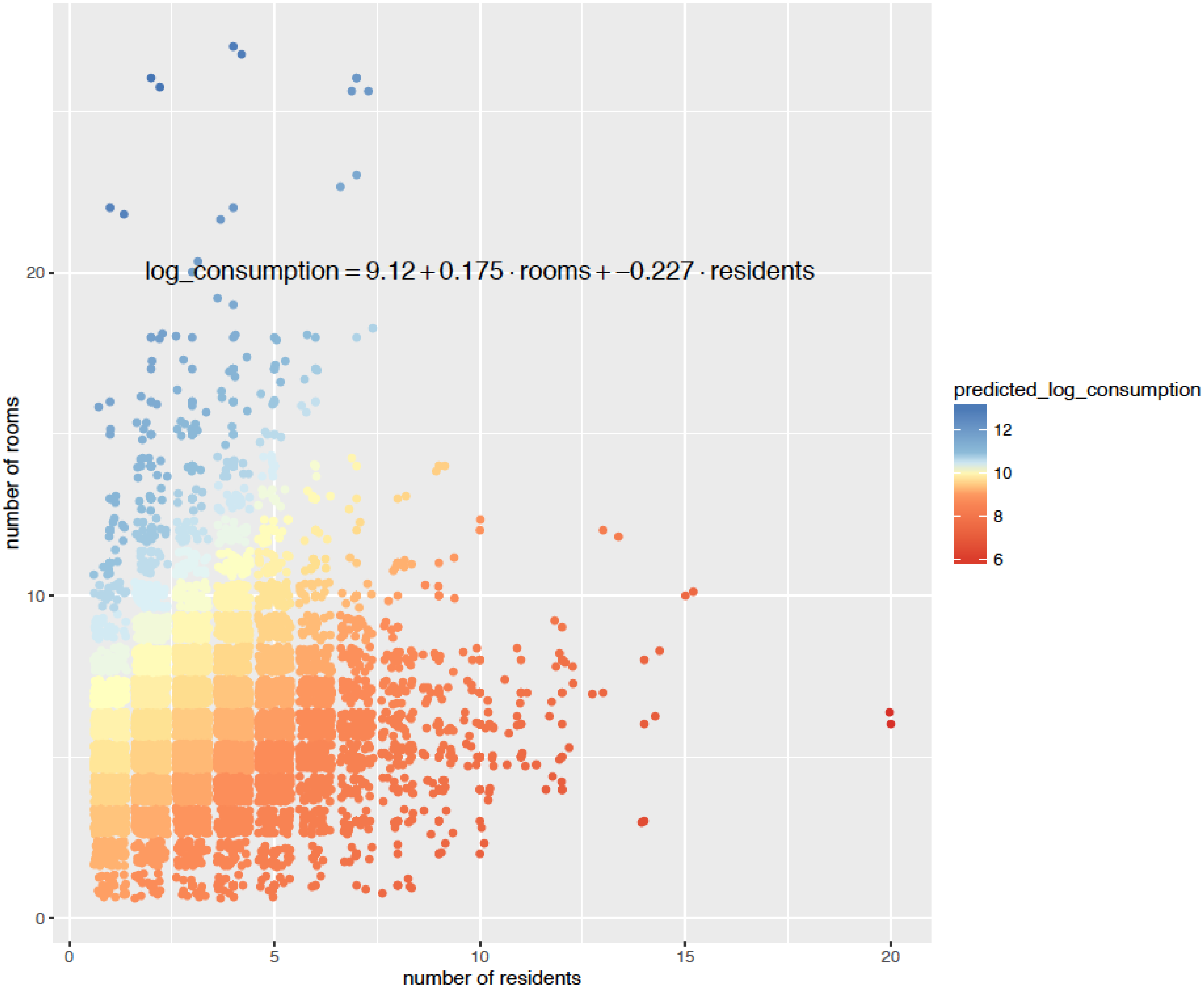


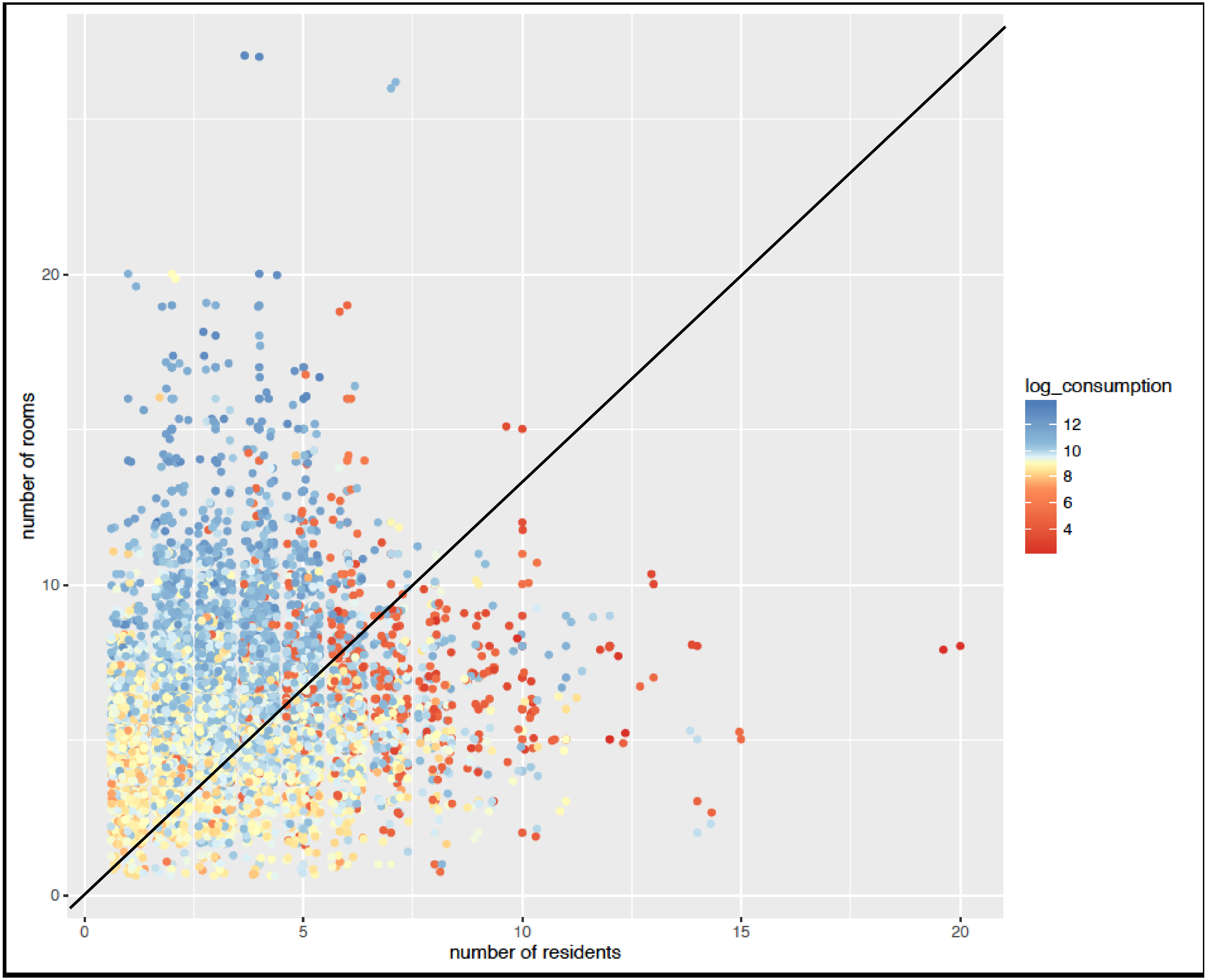


10th Percentile

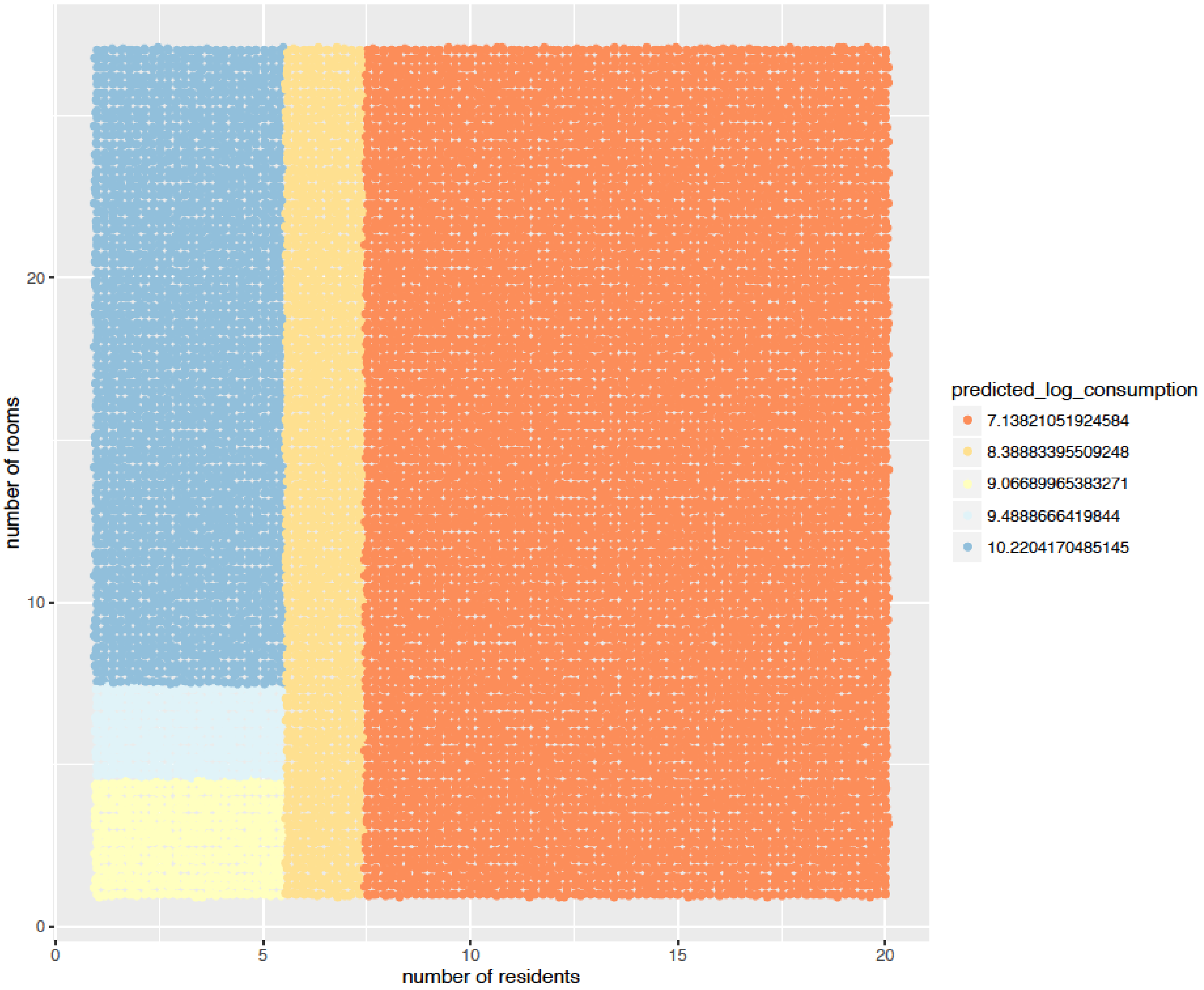

consumption (log)
12.5
10.0
7.5
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
25
26
27
factor(number.of.rooms)

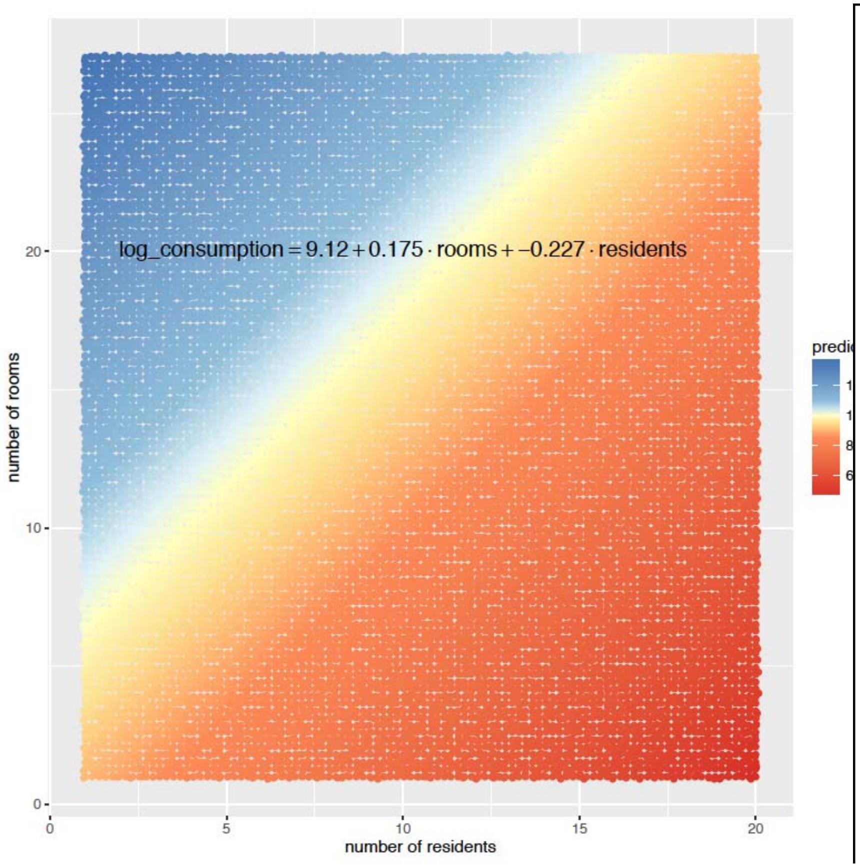

log_consumption = 9.12 + 0.175 · rooms + −0.227 · residents
number of rooms
20
10
0
number of residents
0
5
10
15
20
predicted_log_consumption
12
10
8
6

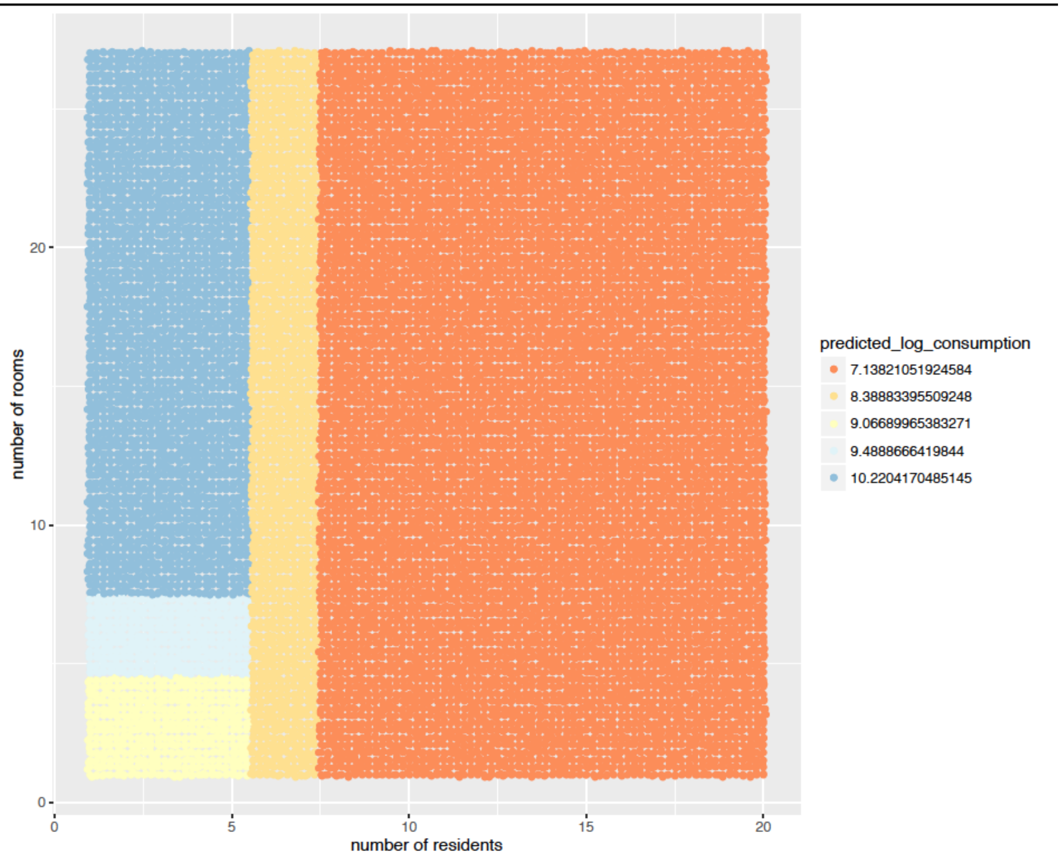

log_consumption = 9.12 + 0.175 · rooms + −0.227 · residents
number of rooms
0
10
20
number of residents
0
5
10
15
20
predicted_log_consumption
12
10
8
6

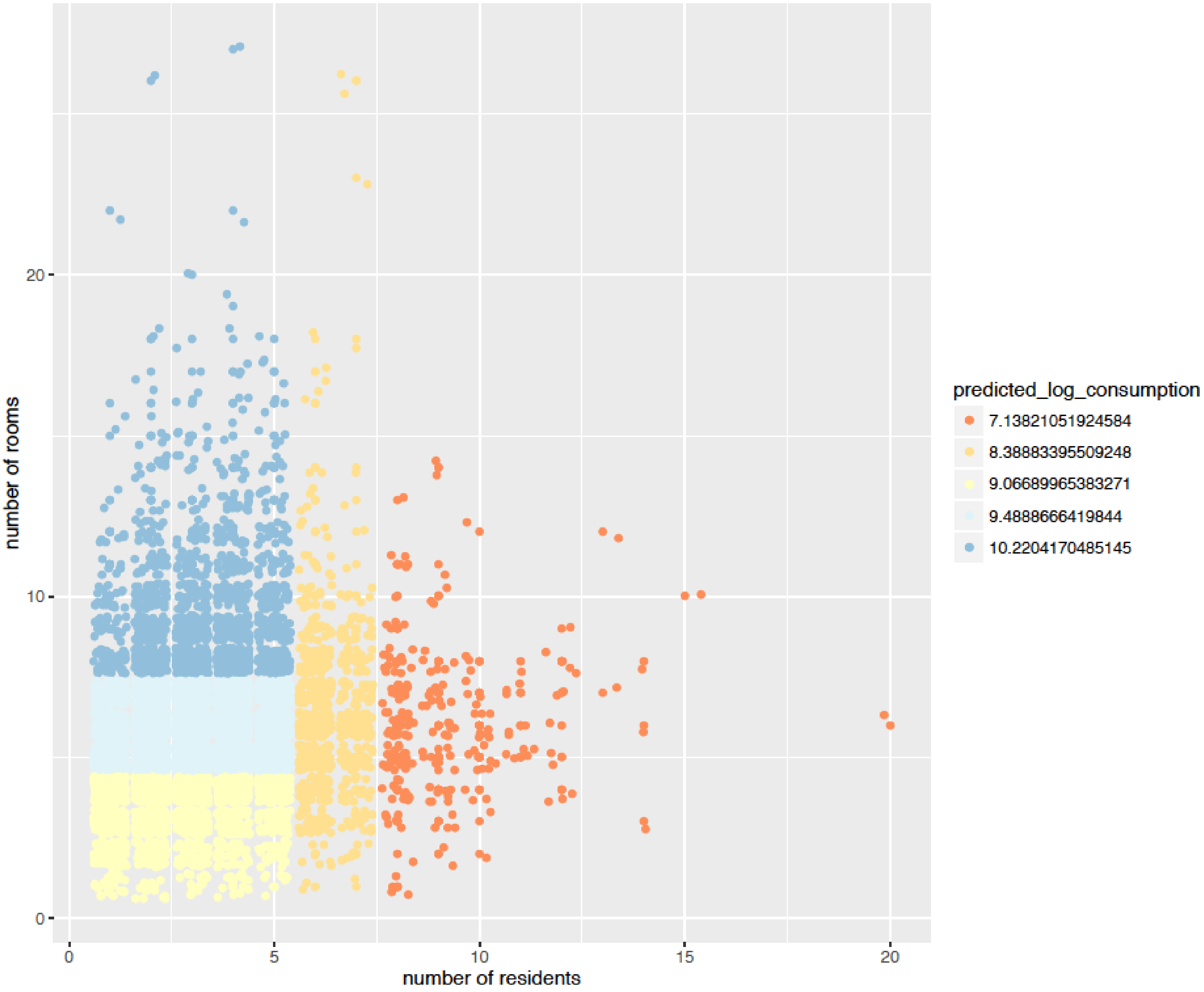

number of rooms
20
10
0
0
5
10
15
20
number of residents
log_consumption
12
10
8
6
4

Two Variable Tree

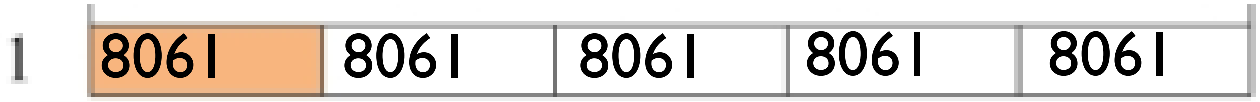



log_consumption = 9.12 + 0.175 · rooms + −0.227 · residents
number of rooms
0
10
20
number of residents
0
5
10
15
20
predi
1
1
8
6


number of rooms
0
10
20
number of residents
0
5
10
15
20
predicted_log_consumption
● 7.13821051924584
● 8.38883395509248
● 9.06689965383271
● 9.4888666419844
● 10.2204170485145


number of rooms
20
10
0
0
5
10
15
20
number of residents
predicted_log_consumption
● 7.13821051924584
● 8.38883395509248
● 9.06689965383271
● 9.4888666419844
● 10.2204170485145

28,573 data points to Fit with



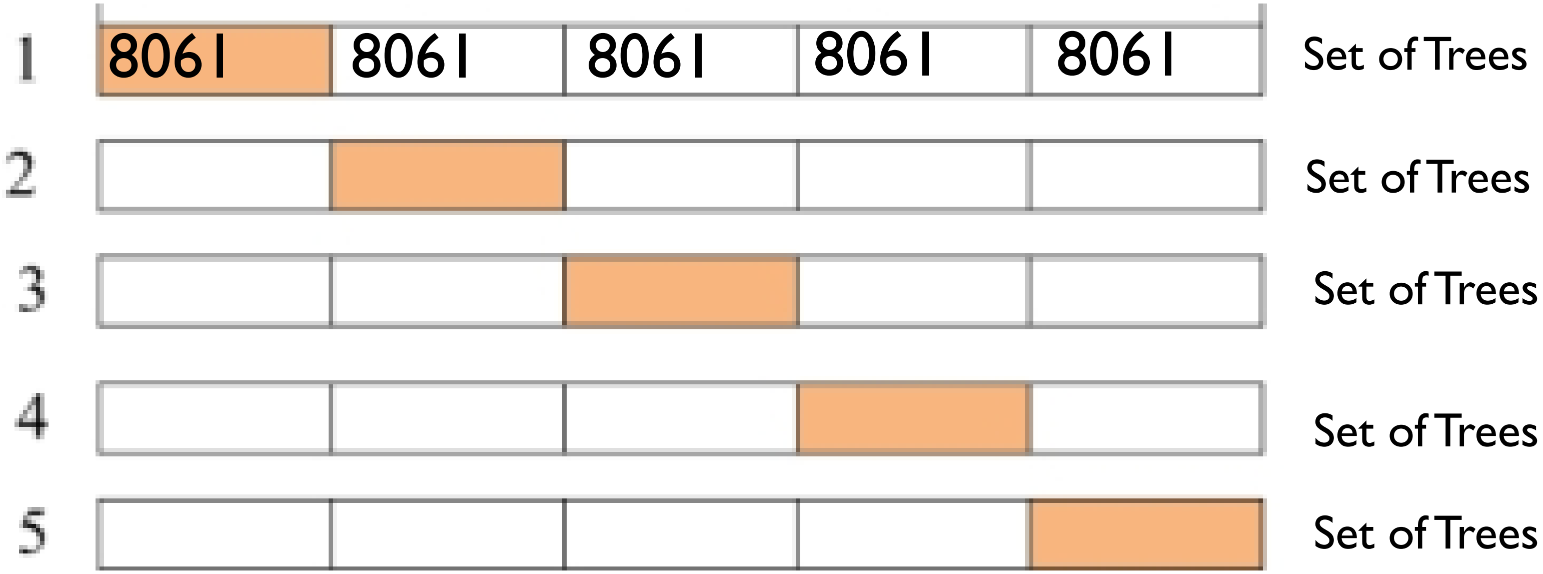
1	8061	8061	8061	8061	8061

Set of Trees

Fit trees on 4/5 of the data

Fit a tree for every level of split size

28,573 data points to Fit with



1 8061 8061 8061 8061 8061 Set of Trees

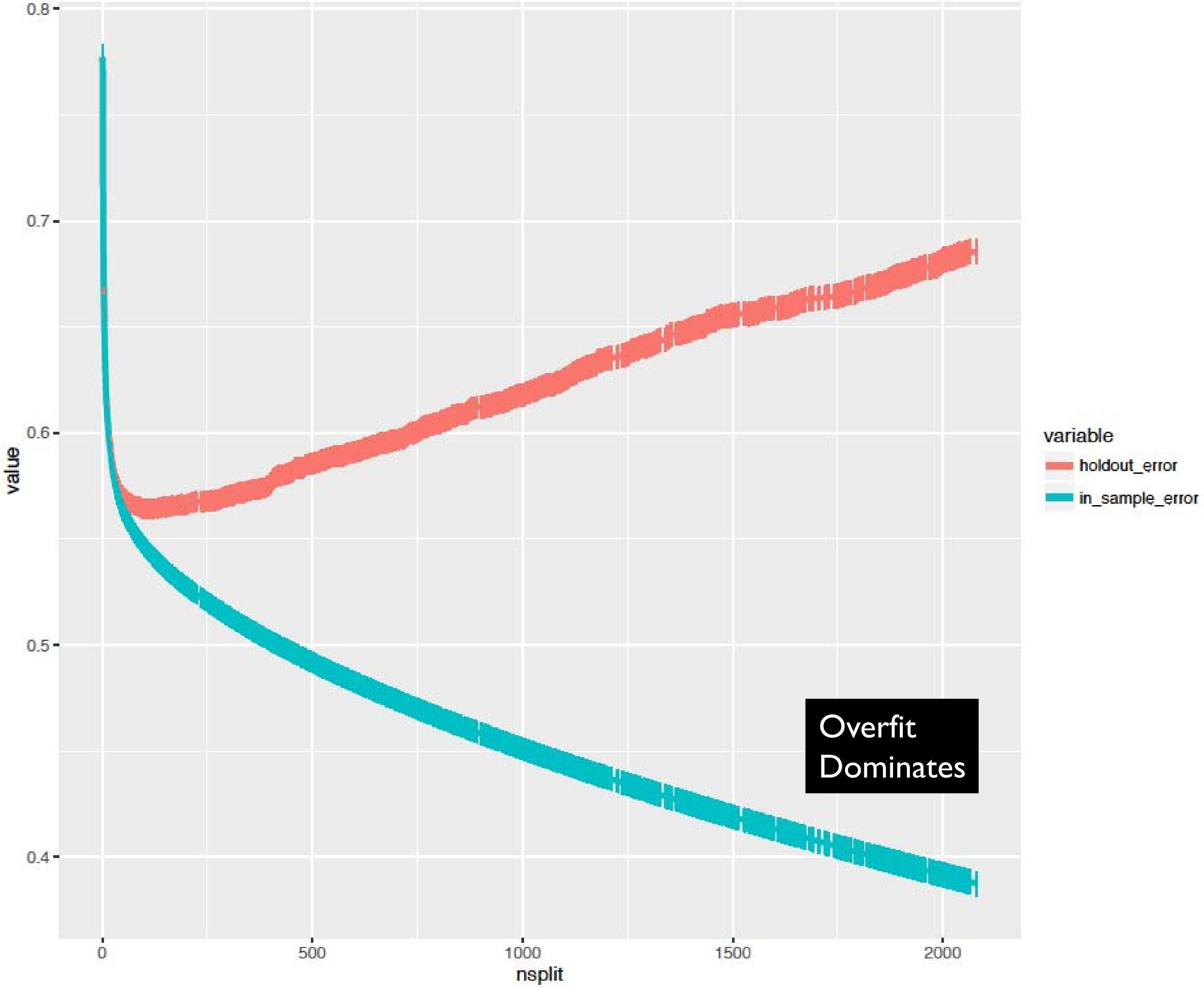
2 Set of Trees

3 Set of Trees

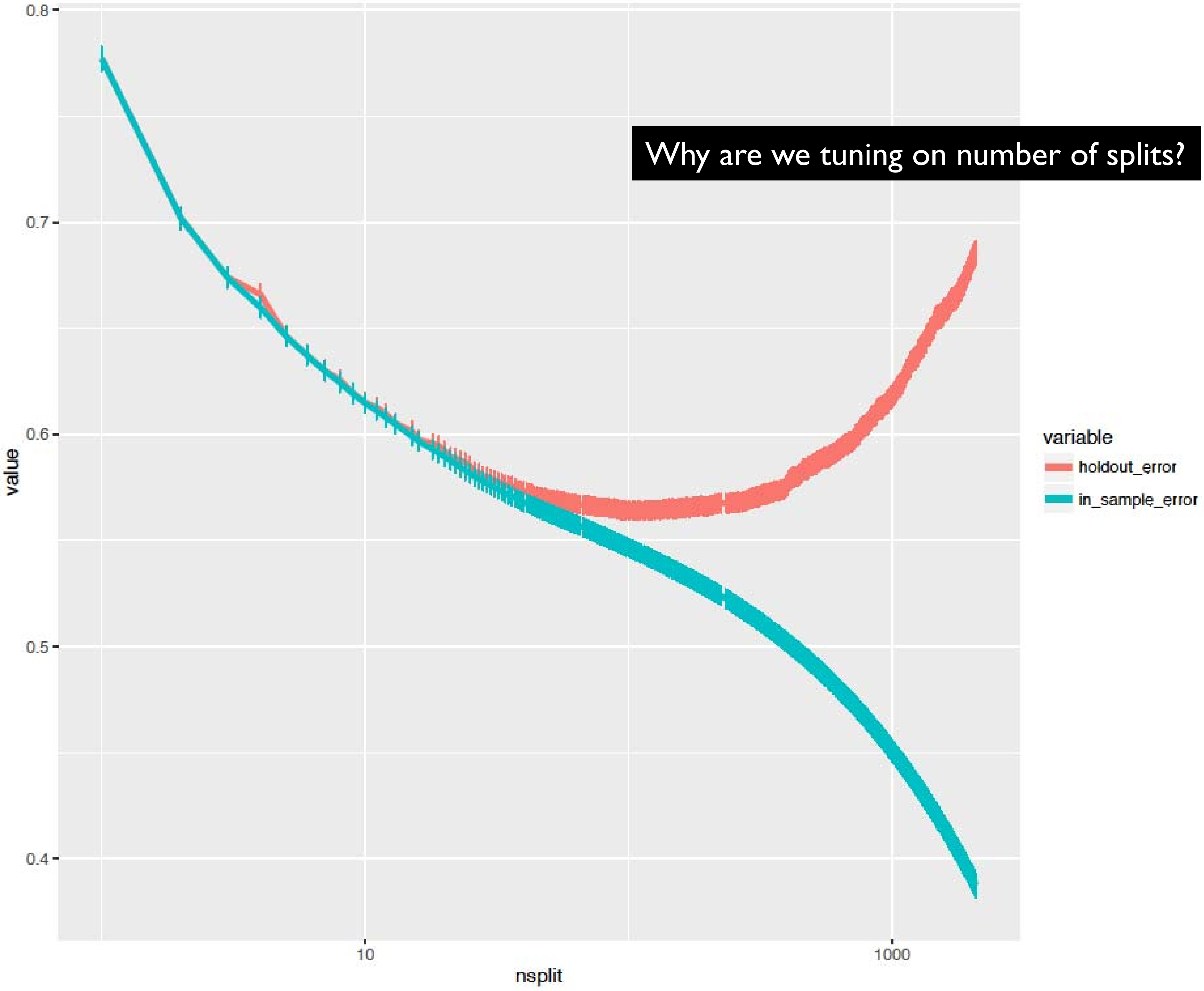
4 Set of Trees

5 Set of Trees

REPEAT leaving each fold out



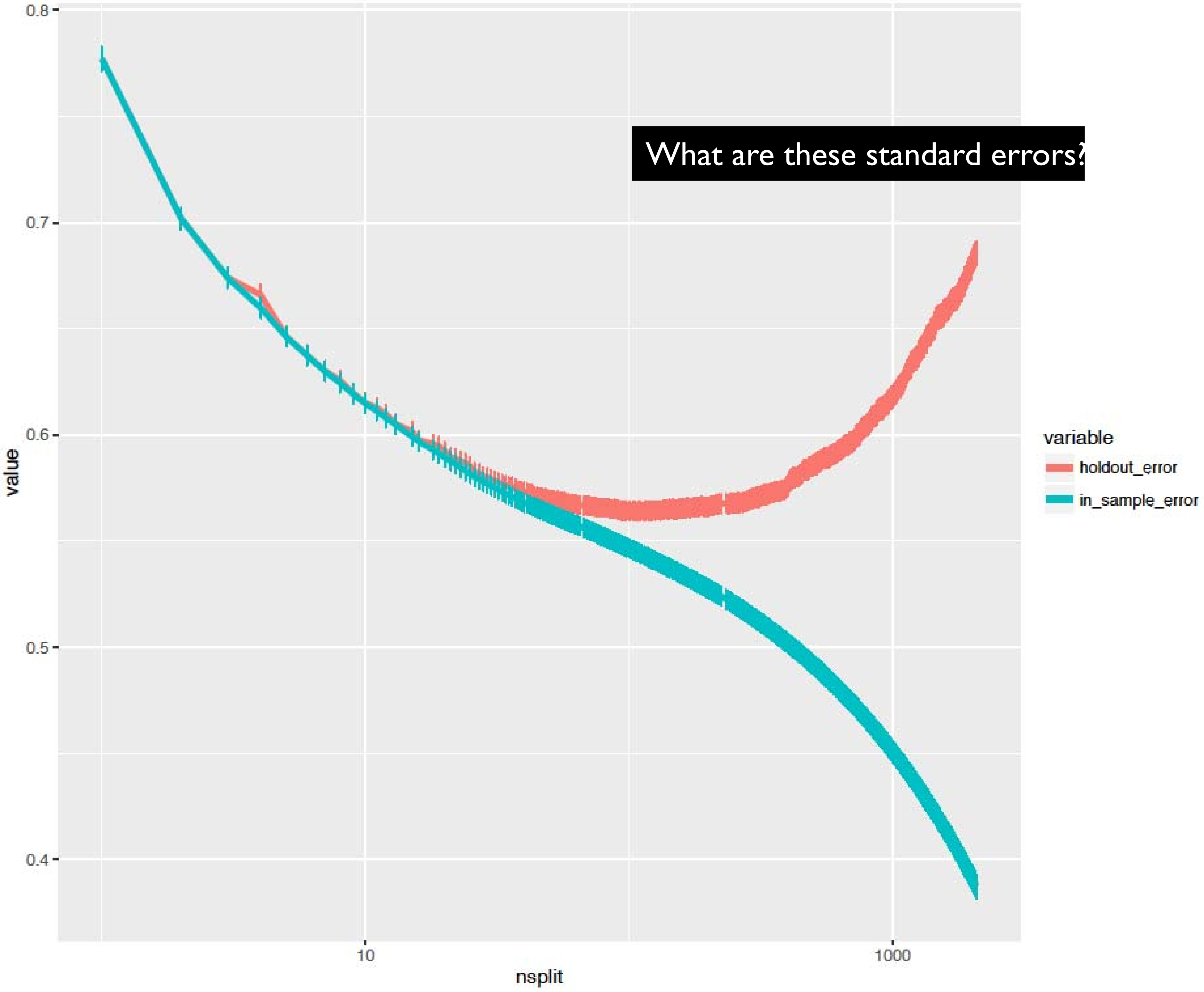
Overfit Dominates



Why are we tuning on number of splits?

Questions and Observations

- How do we choose hold-out set size?
- How to choose the # of folds?
- What to tune on? (regularizer)



What are these standard errors?

Questions and Observations

- How do we choose hold-out set size?
- How to choose the # of folds?
- What to tune on? (regularizer)
- Which tuning parameter to choose from cross-validation?

Tuning Parameter Choice

- Minimum?

- One standard error “rule” (rule of thumb)
  - Which direction?

Output

- Which of these many trees do we output?

- Even after choosing lambda we have as many trees as folds…

- Estimate one tree on full data using chosen cut size

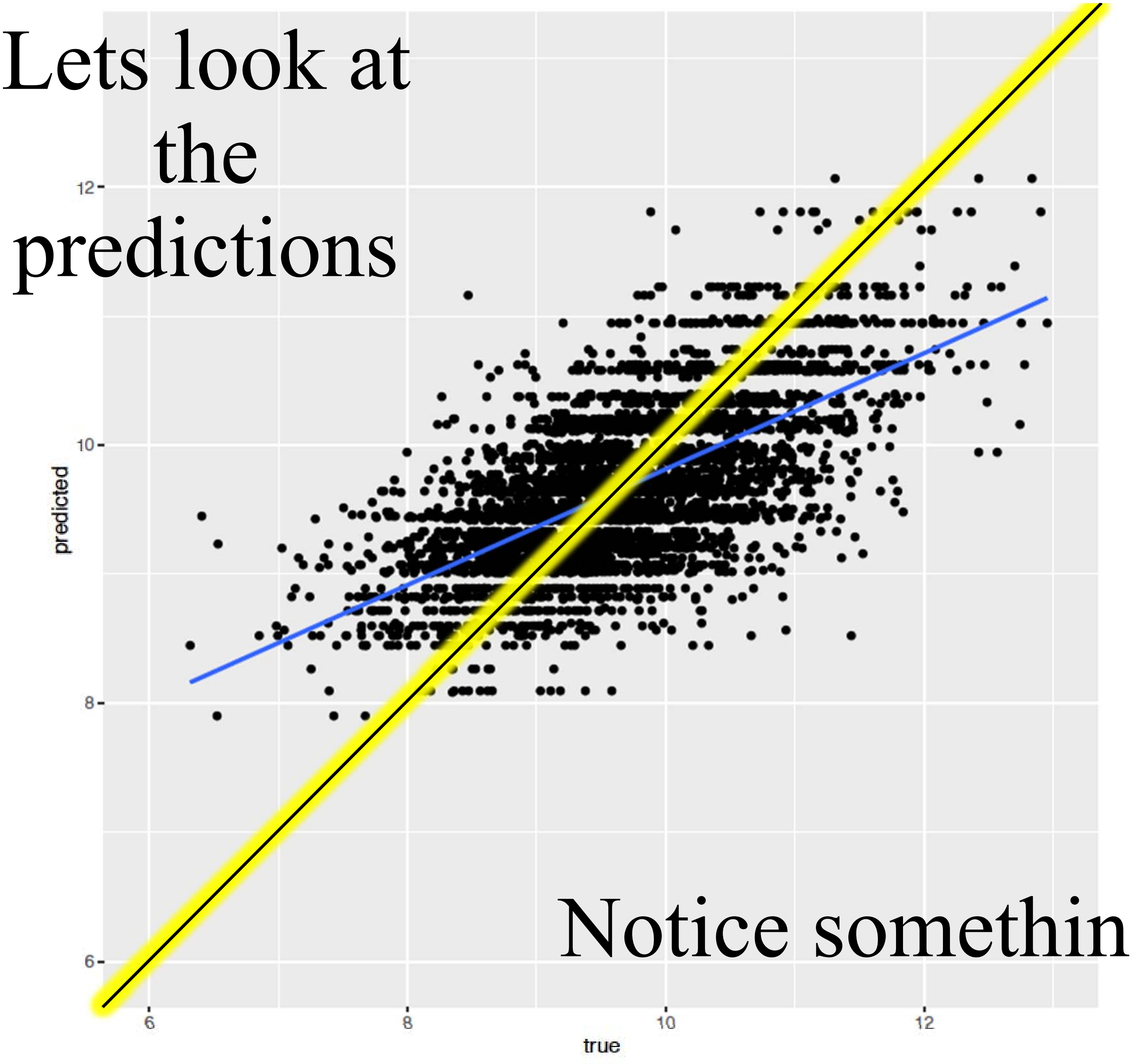
- Key point: Cross validation is just for choosing tuning parameter
  - Just for deciding how complex a model to choose

Questions and Observations

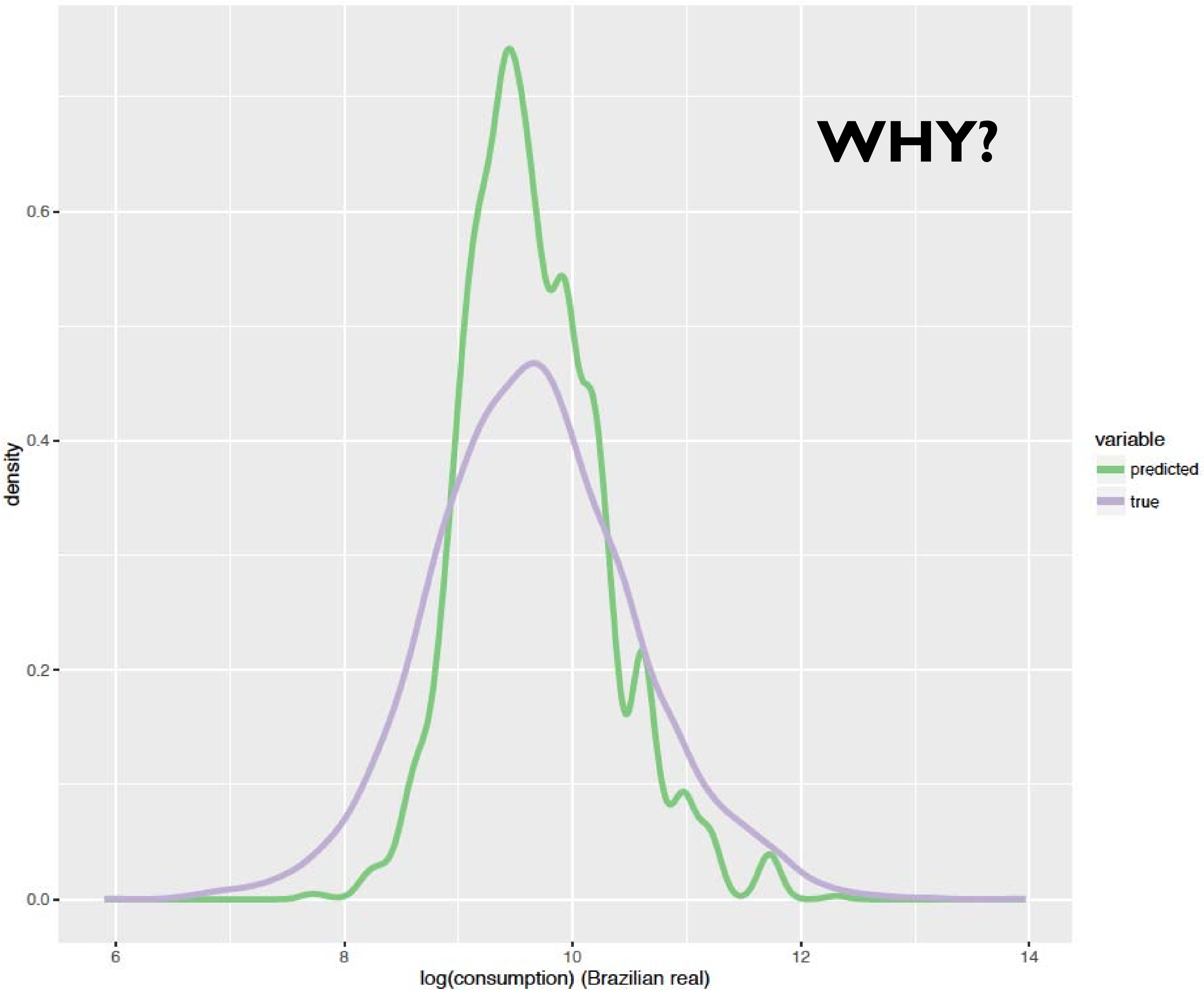
- How do we choose hold-out set size?
- How to choose the # of folds?
- What to tune on? (regularizer)
- Which tuning parameter to choose from cross-validation?

- Is there a problem tuning on subsets and then outputting fitted value on full set?

Lets look at the predictions

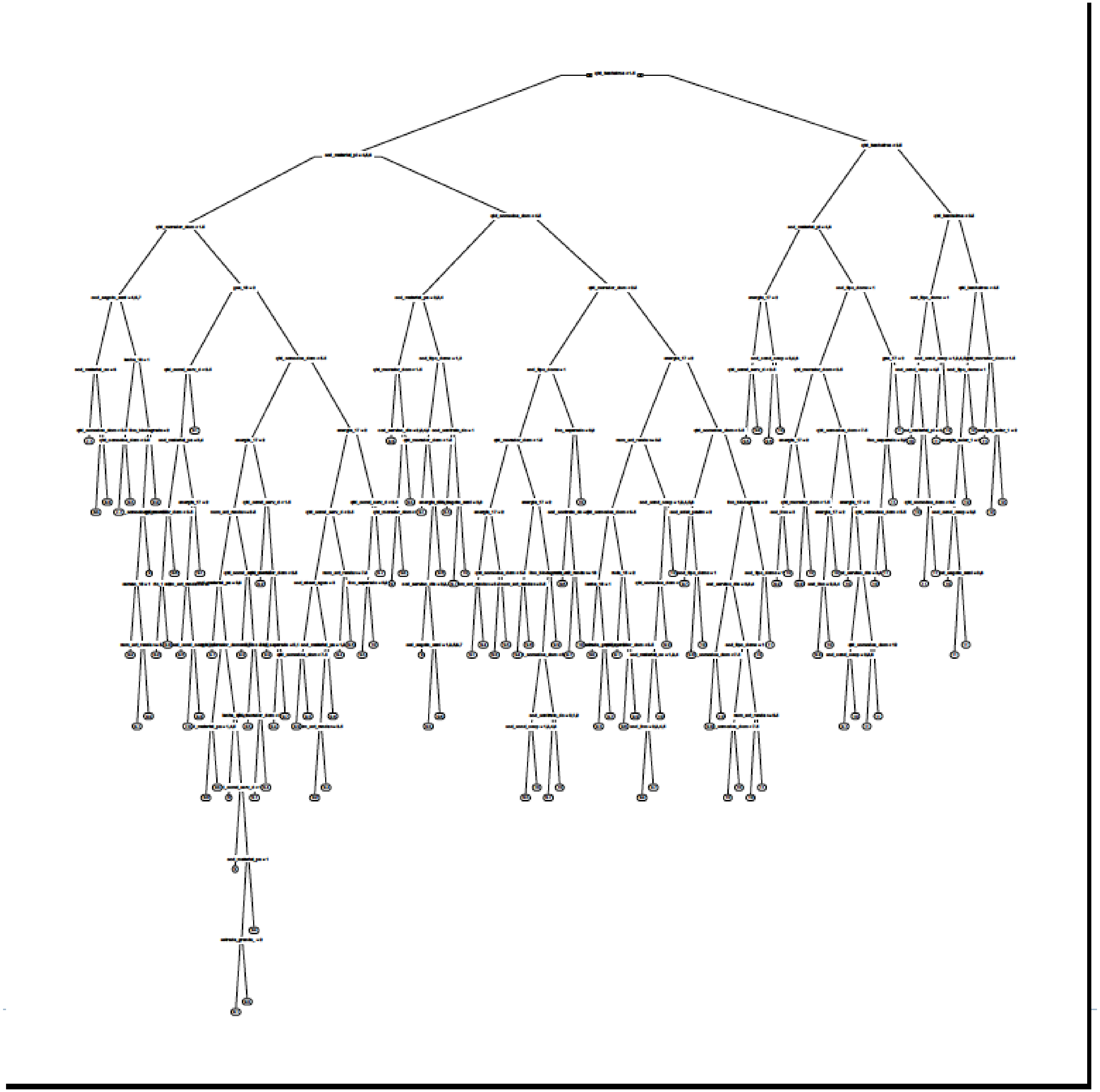


Notice something?


0.0
0.2
0.4
0.6
density
6
8
10
12
14
log(consumption) (Brazilian real)
variable
predicted
true

WHY?

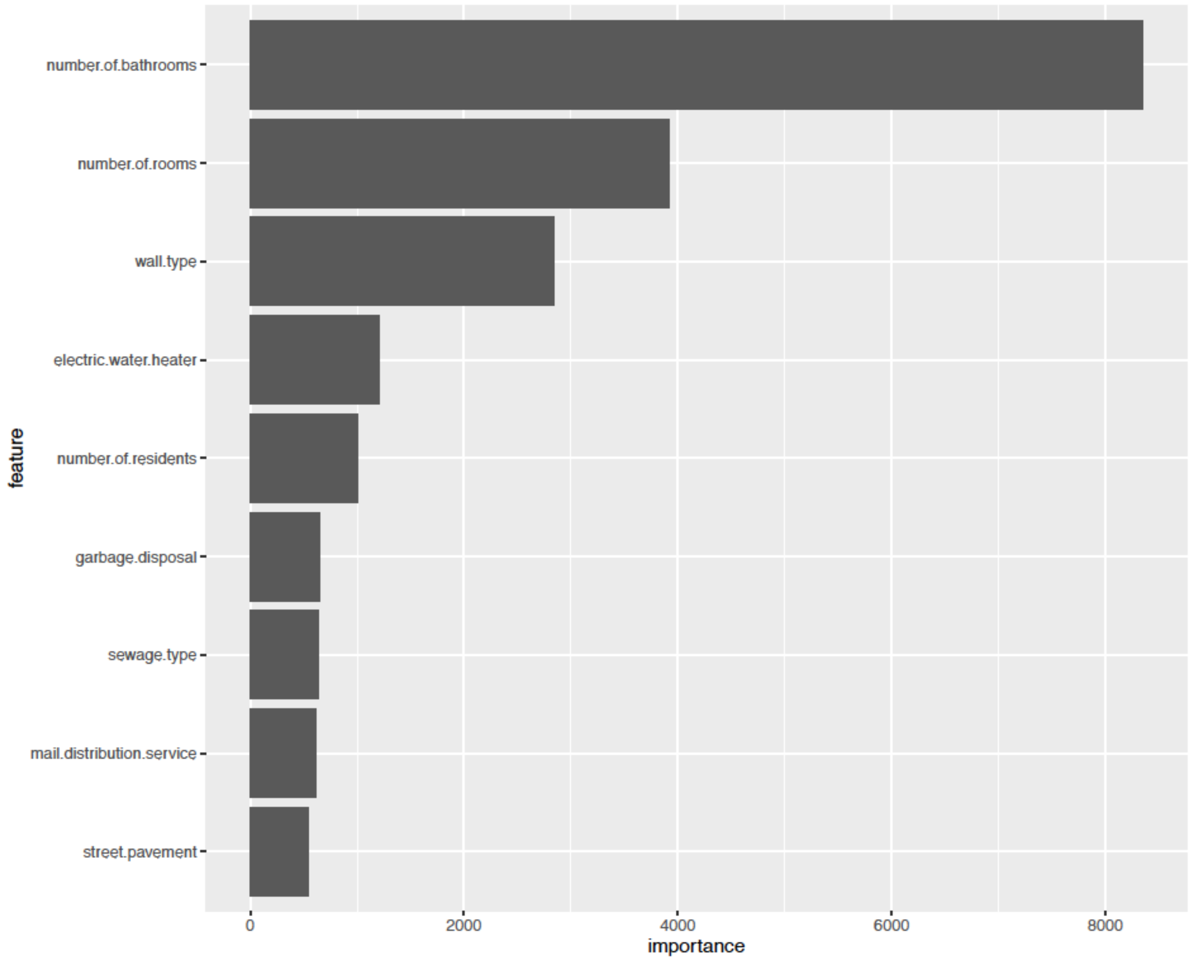
What does the tree look like?



- What else can we look at to get a sense of what the predictions are?

Variable Importance

Empirical loss by noising up *x* minus Empirical loss



How to describe model

- Large discussion of "interpretability"
  - Will return to this

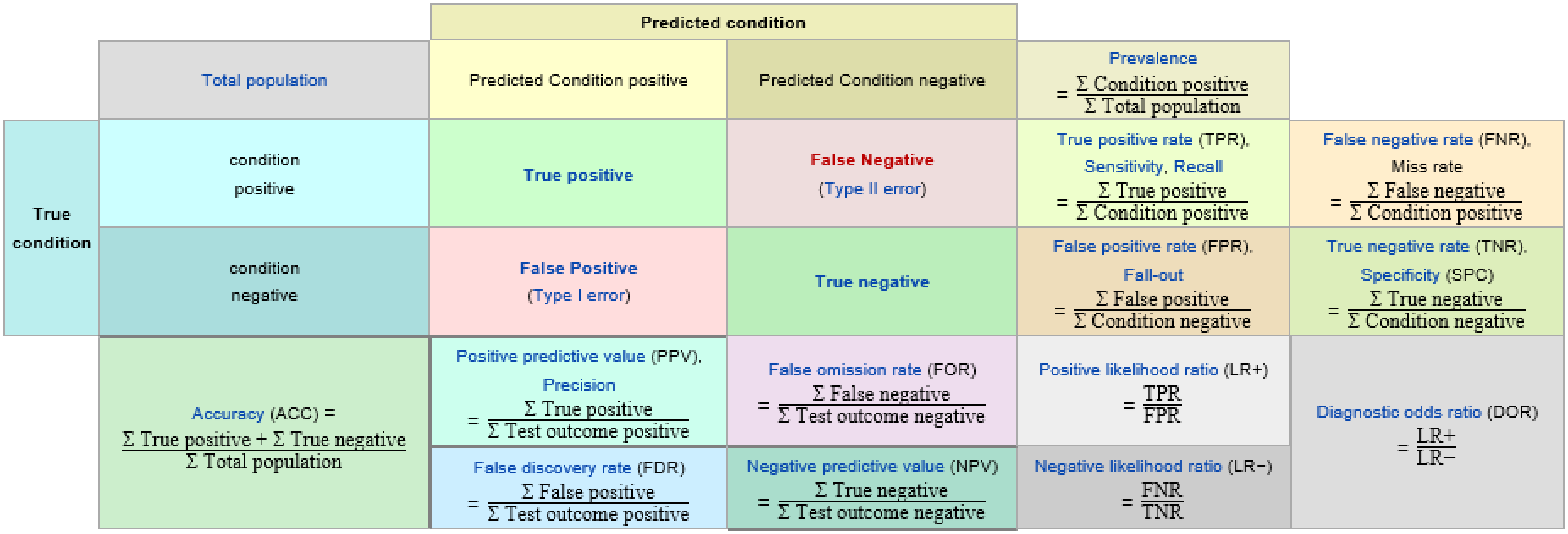
- But one implication is that the prediction function itself becomes a new *y* variable to analyze.

- Is any of this stable? What would a confidence interval look like?

Questions and Observations

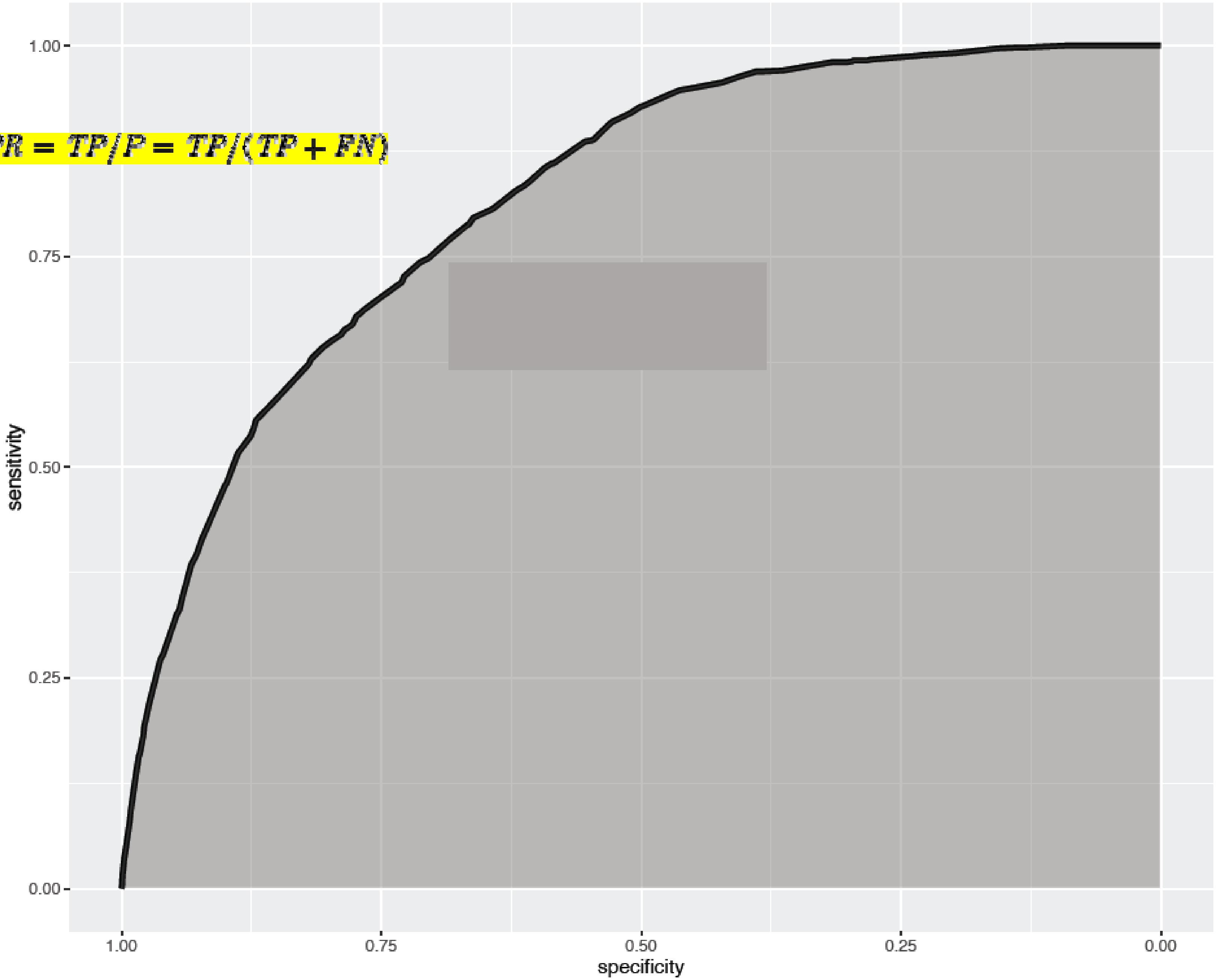
- How do we choose hold-out set size?
- How to choose the # of folds?
- What to tune on? (regularizer)
- Which tuning parameter to choose from cross-validation?
- Is there a problem tuning on subsets and then outputting fitted value on full set?
- What is stable/robust about the estimated function?

Measuring Performance



		Predicted condition			
	Total population	Predicted Condition positive	Predicted Condition negative	Prevalence $= \frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$	
True condition	condition positive	**True positive**	**False Negative** (Type II error)	True positive rate (TPR), Sensitivity, Recall $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False negative rate (FNR), Miss rate $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$
	condition negative	**False Positive** (Type I error)	**True negative**	False positive rate (FPR), Fall-out $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	True negative rate (TNR), Specificity (SPC) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$
	Accuracy (ACC) = $\frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$	Positive predictive value (PPV), Precision $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Test outcome positive}}$	False omission rate (FOR) $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Test outcome negative}}$	Positive likelihood ratio (LR+) $= \frac{\text{TPR}}{\text{FPR}}$	Diagnostic odds ratio (DOR) $= \frac{\text{LR+}}{\text{LR−}}$
		False discovery rate (FDR) $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Test outcome positive}}$	Negative predictive value (NPV) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Test outcome negative}}$	Negative likelihood ratio (LR−) $= \frac{\text{FNR}}{\text{TNR}}$	

$TPR = TP/P = TP/(TP + FN)$

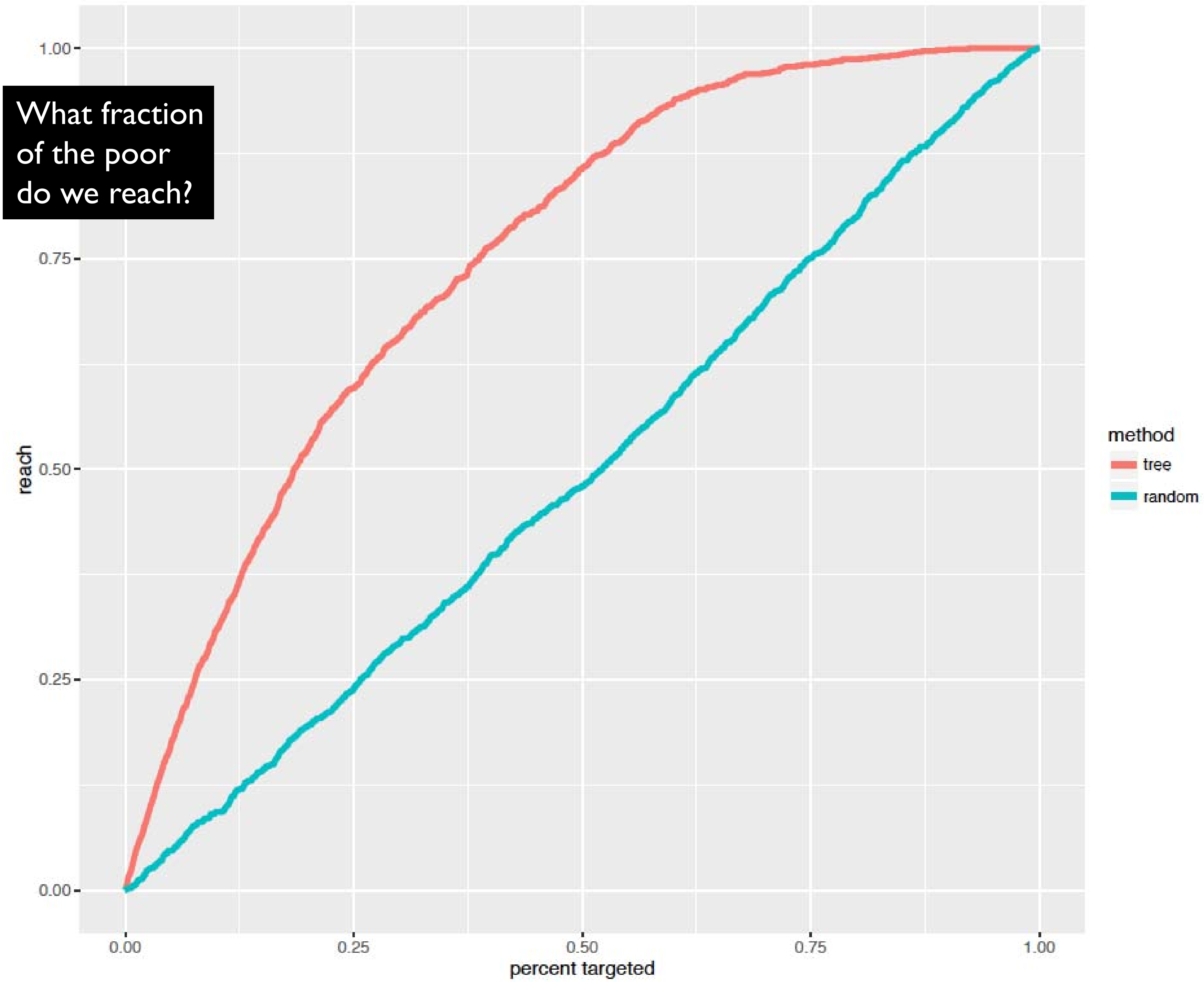


$SPC = TN/N = TN/(TN + FP)$

Measuring Performance

- Area Under Curve: Typical measure of performance

- What do you think of this measure?



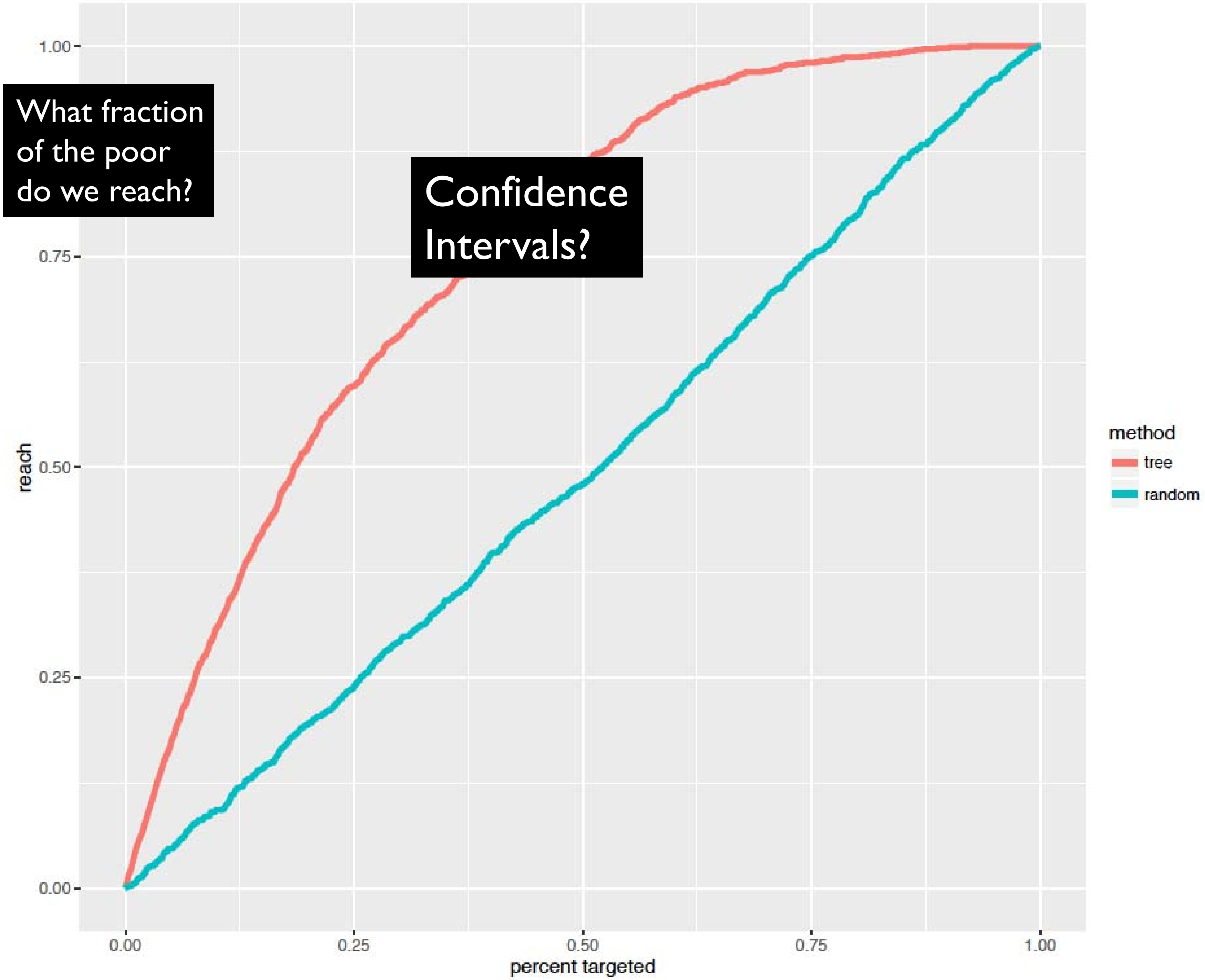
What fraction of the poor do we reach?

Measuring Performance

- AUC: Typical measure of performance

- What do you think of this measure?

- Getting the domain specific meaningful performance measure
  - **Magnitudes**
  - **Need point of comparison**



What fraction of the poor do we reach?

Confidence Intervals?

This is what we want from econometric theorems

- How do we choose hold-out set size?
- How to choose the # of folds?
- What to tune on? (regularizer)
- Which tuning parameter to choose from cross-validation?
- Is there a problem tuning on subsets and then outputting fitted value on full set?
- What is stable/robust about the estimated function?
- How do we form standard errors on performance?

Summary

- Regression trees easy to understand and interpret
- Tradeoff between personalized versus inaccurate predictions
- Cross-validation is a tool to figure out the best balance in a particular dataset
  - E.g if truth is complex, may want to go deeper

- CART is ad hoc, but works well in practice
  - Loses to OLS/logit if true model is linear
  - Good at finding lots of complex interactions