Bayesian Boosting

or how I learned to stop worrying and love uncertainty

George Perrett

2023-07-14

A Choice

For an outcome of interest we want higher values.

Option A

With option A our model predicts a value of 9

Option B

With option B our model predicts a value of 9

A Choice

For an outcome of interest we want higher values.

Option A

With option A our model predicts a value of 9

Option B

With option B our model predicts a value of 9

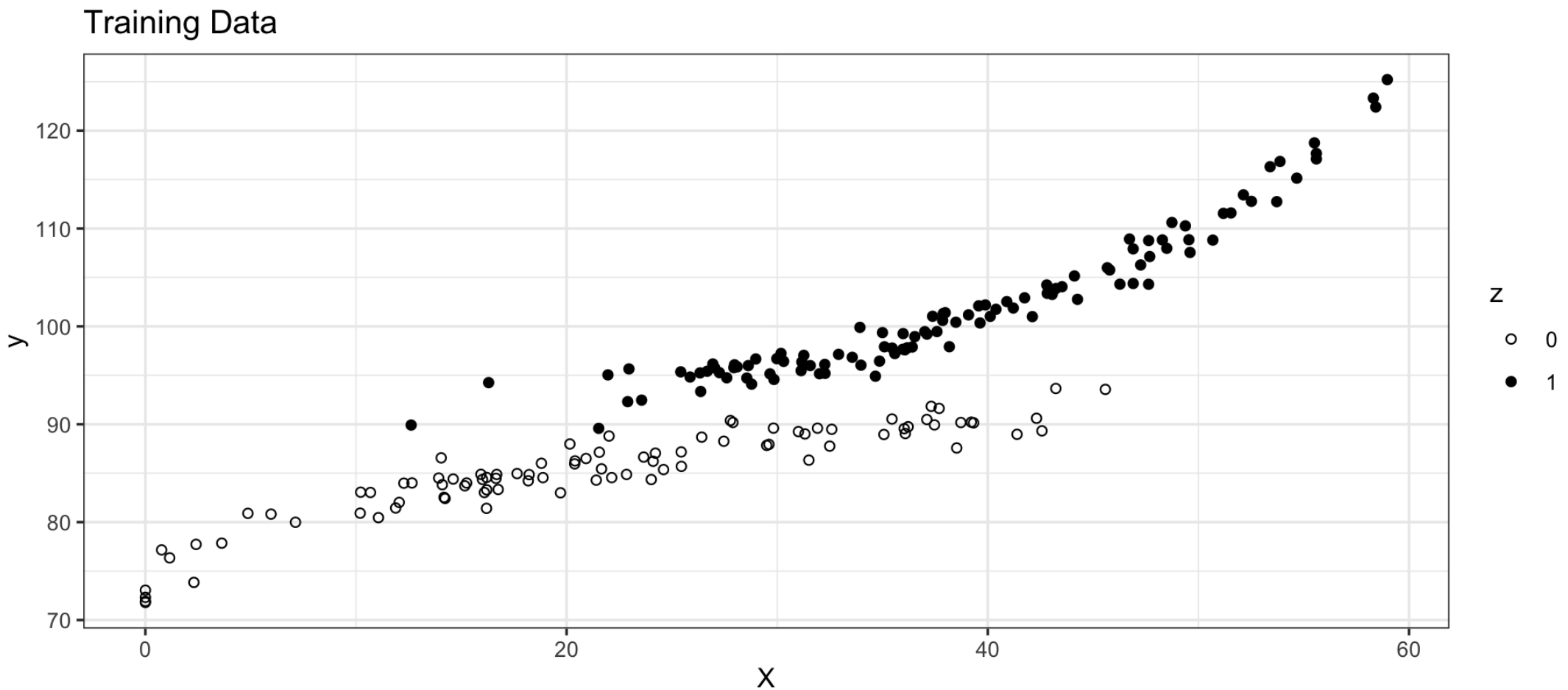
Option A

Option A is predicted to have a 95% chance of being between -3 and 11

Option B

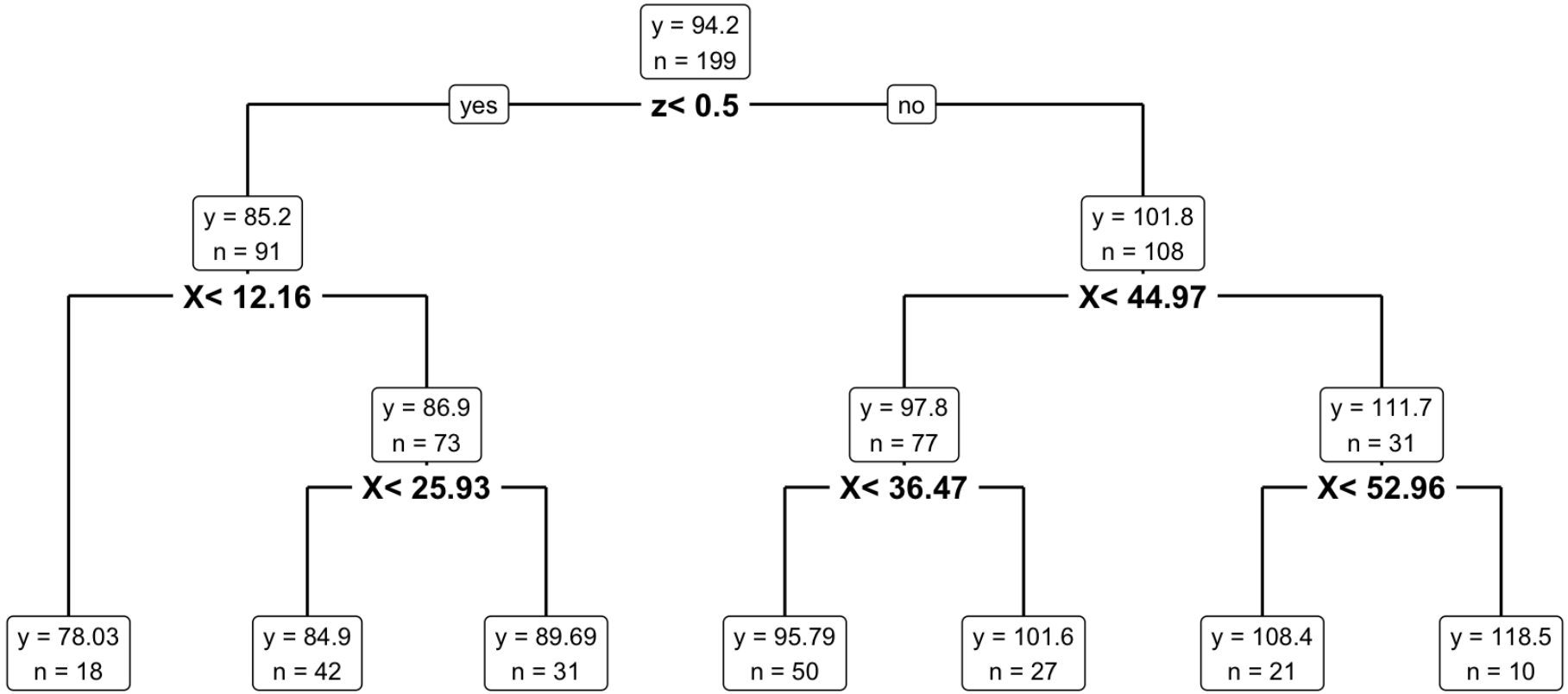
Option B is predicted to have a 95% chance of being between 8 and 10

Training Data

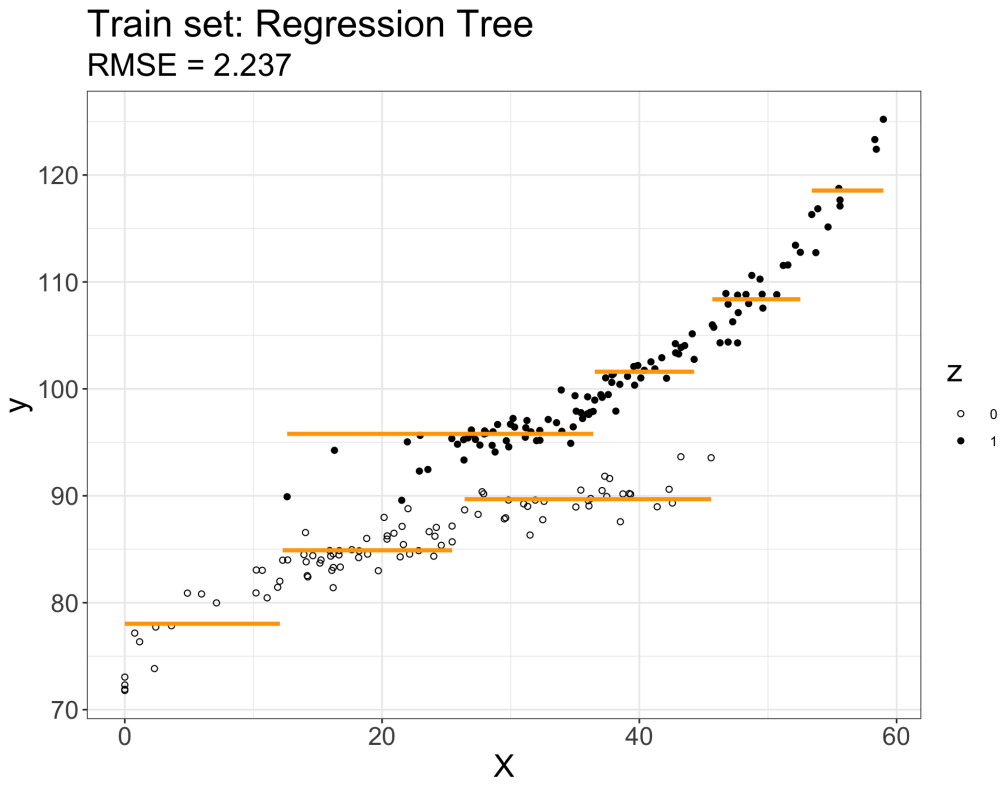


Regression Tree

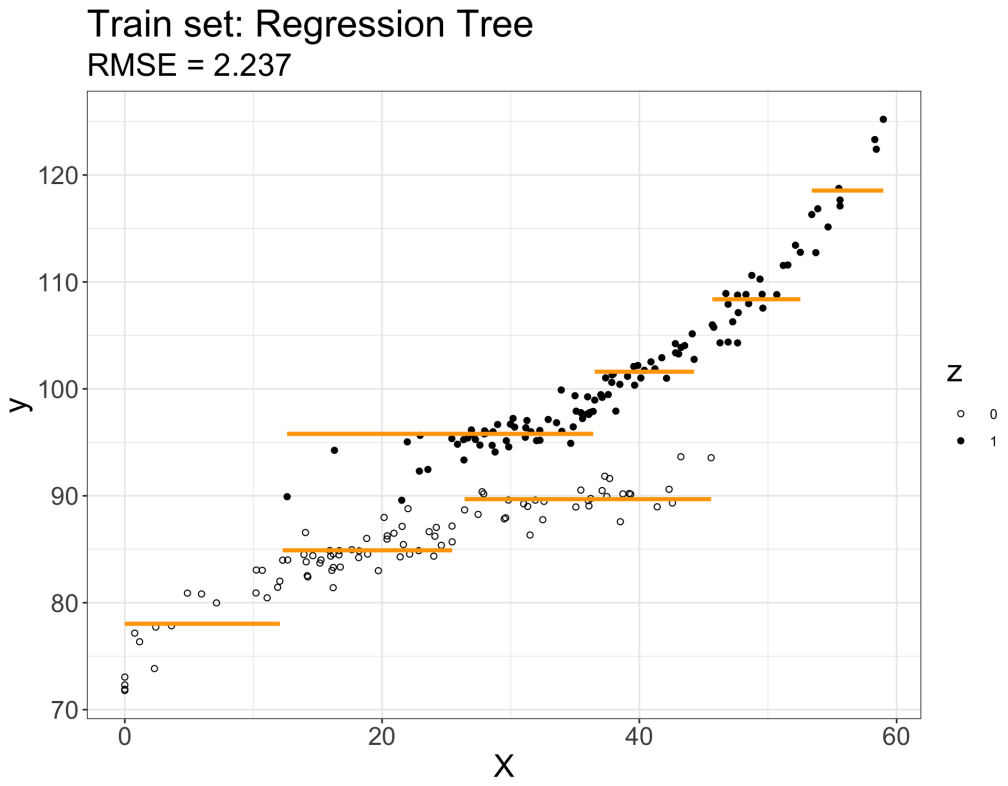
Regression Trees are the bulding blocks of non-Bayesian and Bayesian Boosting

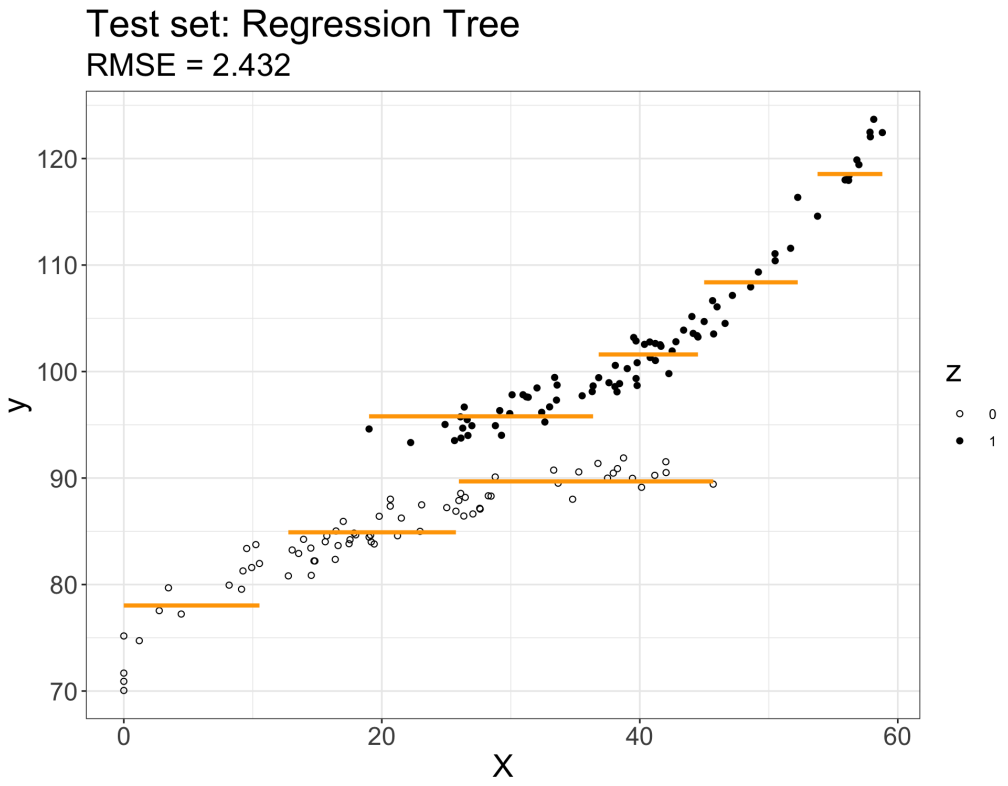


Regression Tree Results



Regression Tree Results





Gradient Boosted Regression Trees the Logic

By combining many small regression trees, we can improve our predictions.

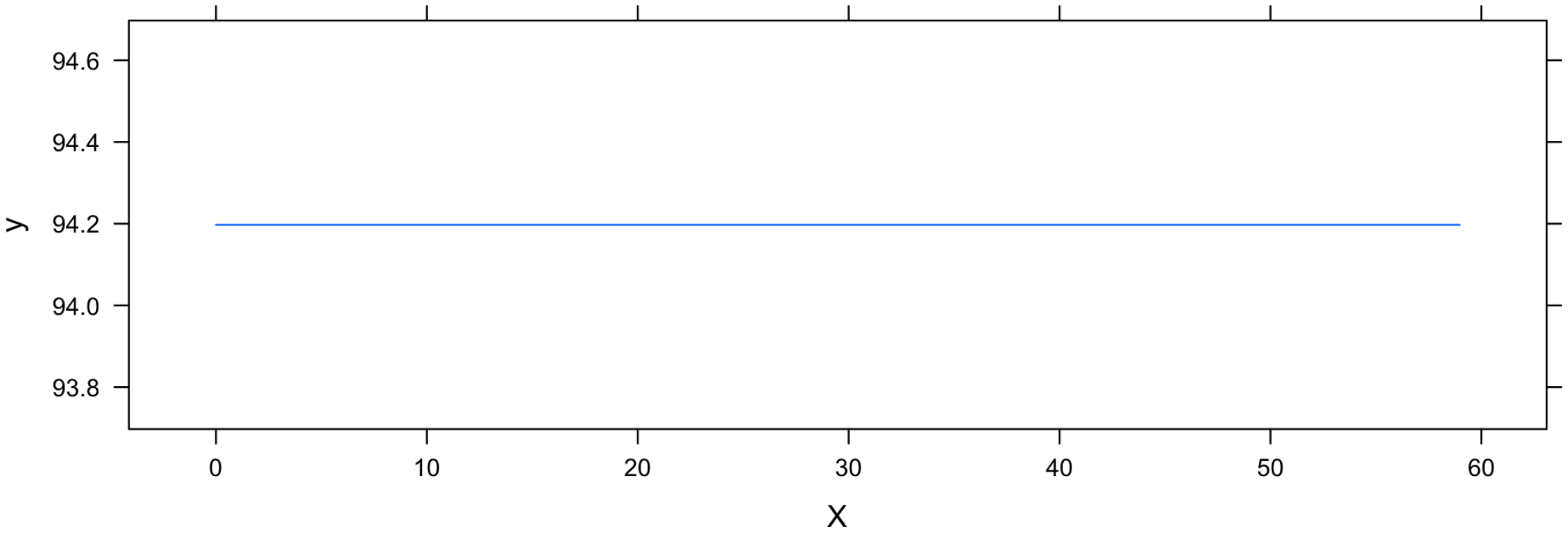
The idea of Gradient Boosted Regression Trees is to add the contributions of each weak tree together. Trees are "boosted" by previous weak trees.

Trees are kept weak by setting the number of trees, the depth of the trees and a shrinkage parameter that is multiplied to the fit of each tree.

These are all tunable parameters.

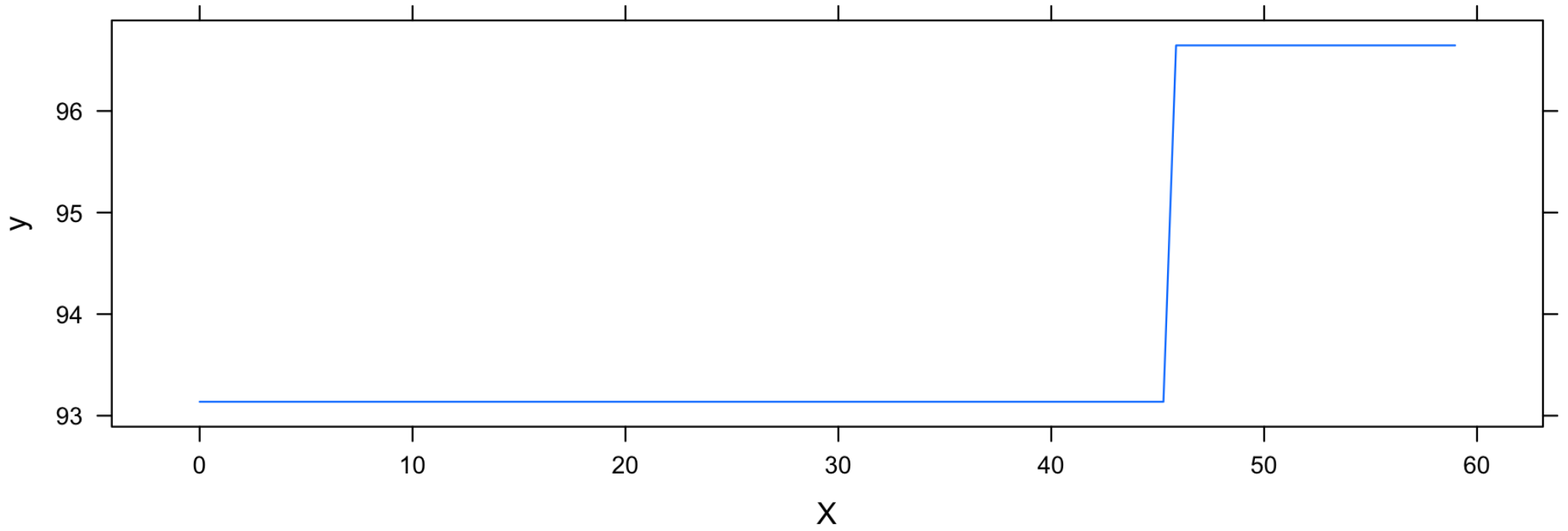
Gradient Boosted Regression Trees: Step 1

Use the mean outcome as the prediction.



Gradient Boosted Regression Trees: Step 2

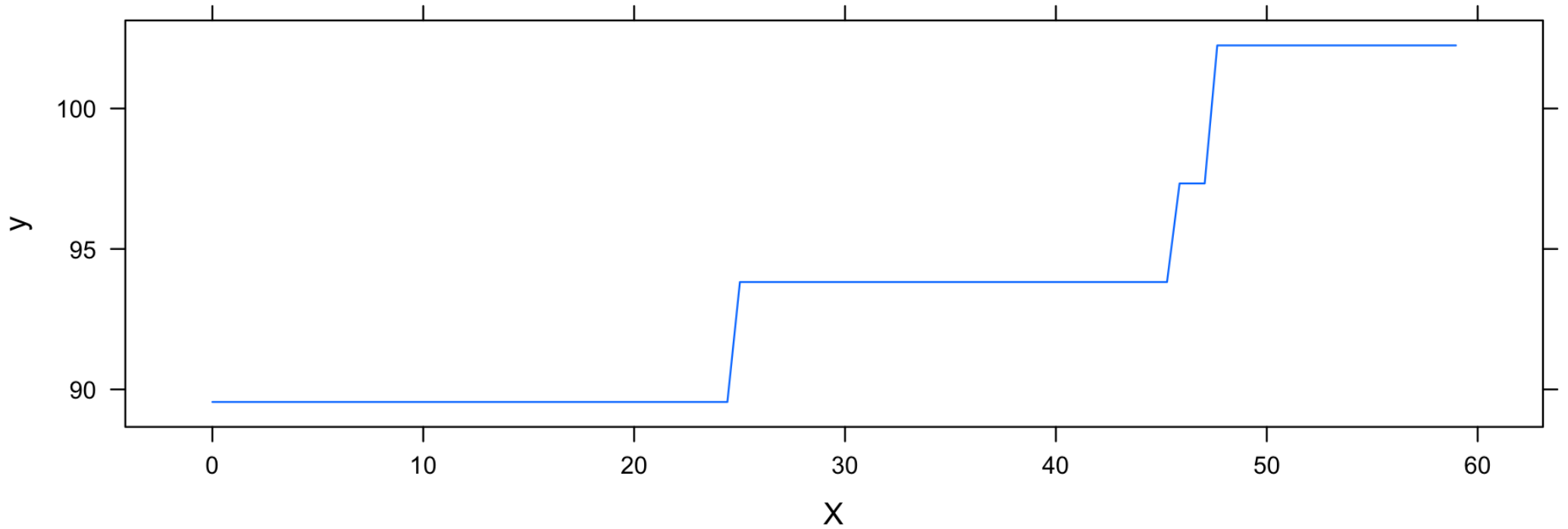
Take the residual from the mean and fit a weak regression tree multiplied by the shinkage parameter.



Gradient Boosted Regression Trees: Step 3

Take the residuals from the last fit and fit another weak tree!

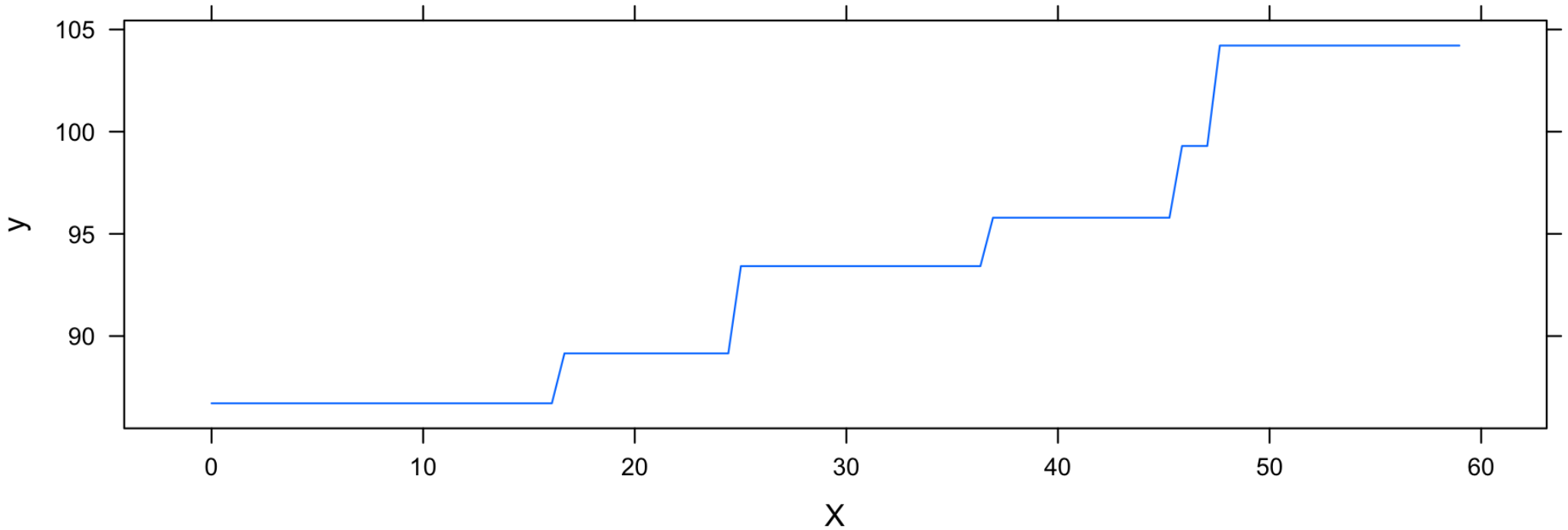
Multiply this tree by the shrinkage parameter and add all the trees together!



Gradient Boosted Regression Trees: Step 4

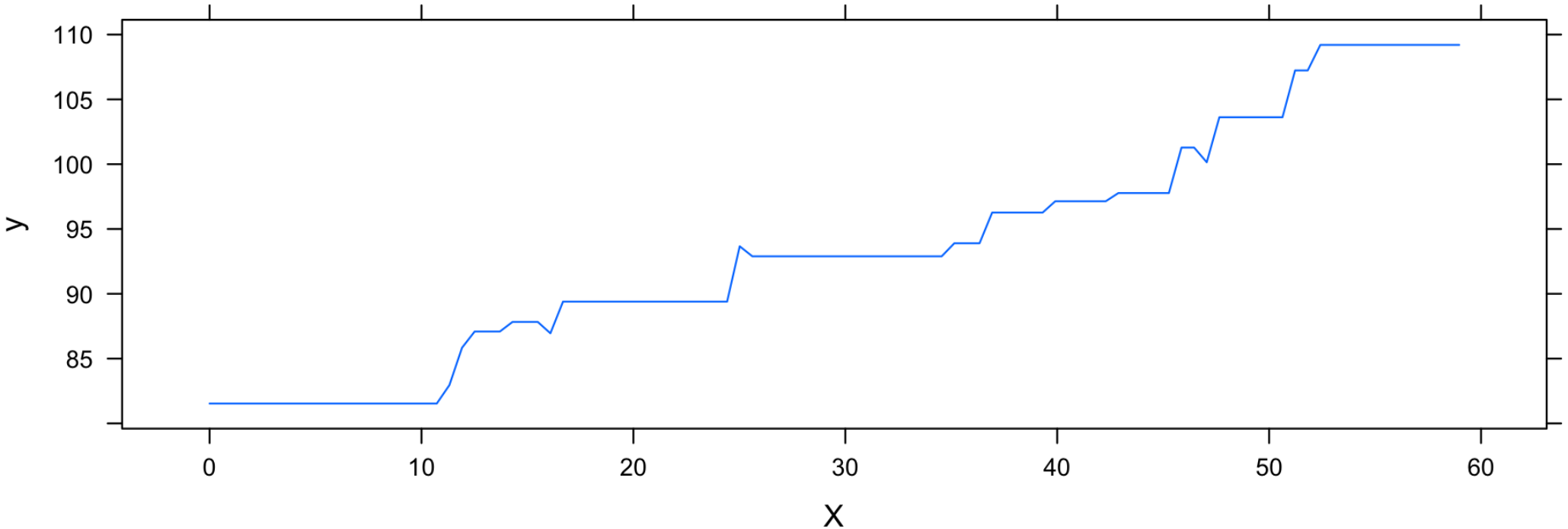
Repeat!

Fit from the third tree:

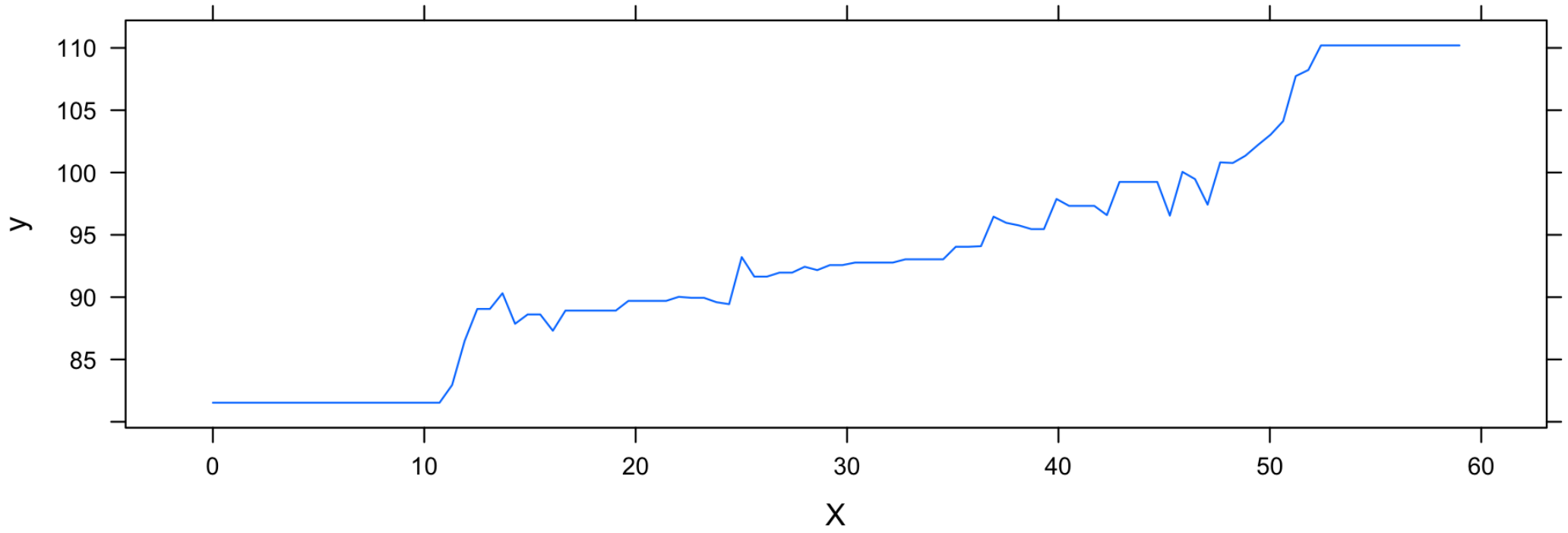


Gradient Boosted Regression Trees: Tree 10

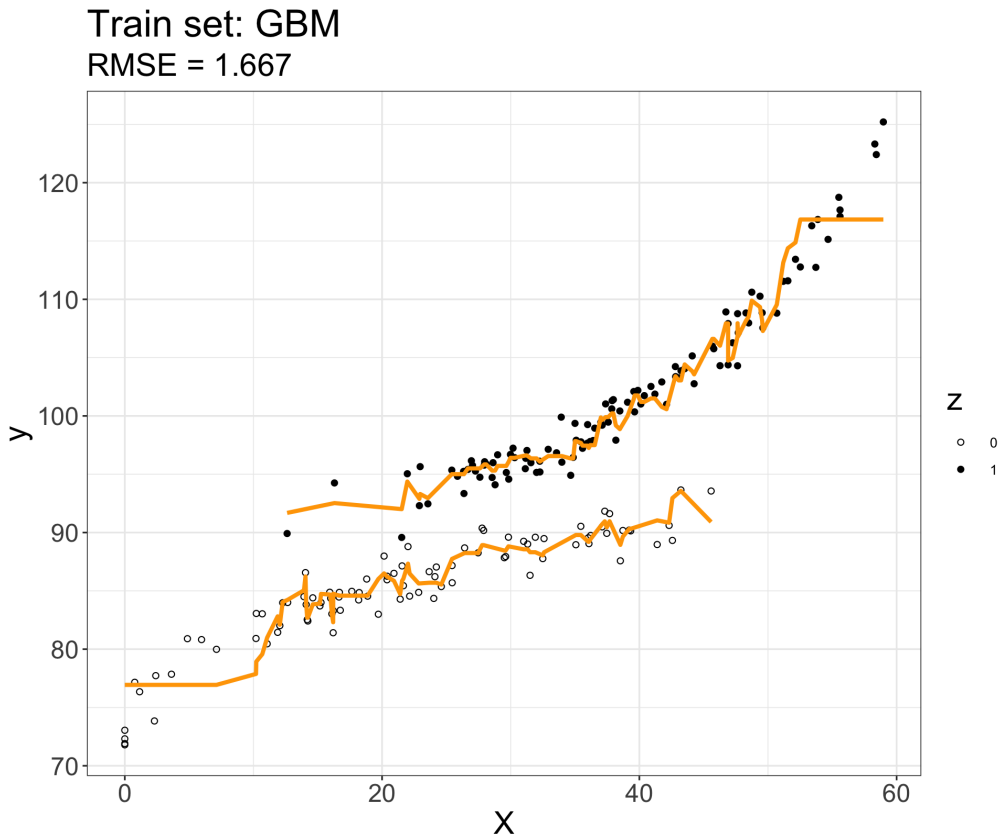
The fit slowly changes, this slow fitting prevents overfitting.



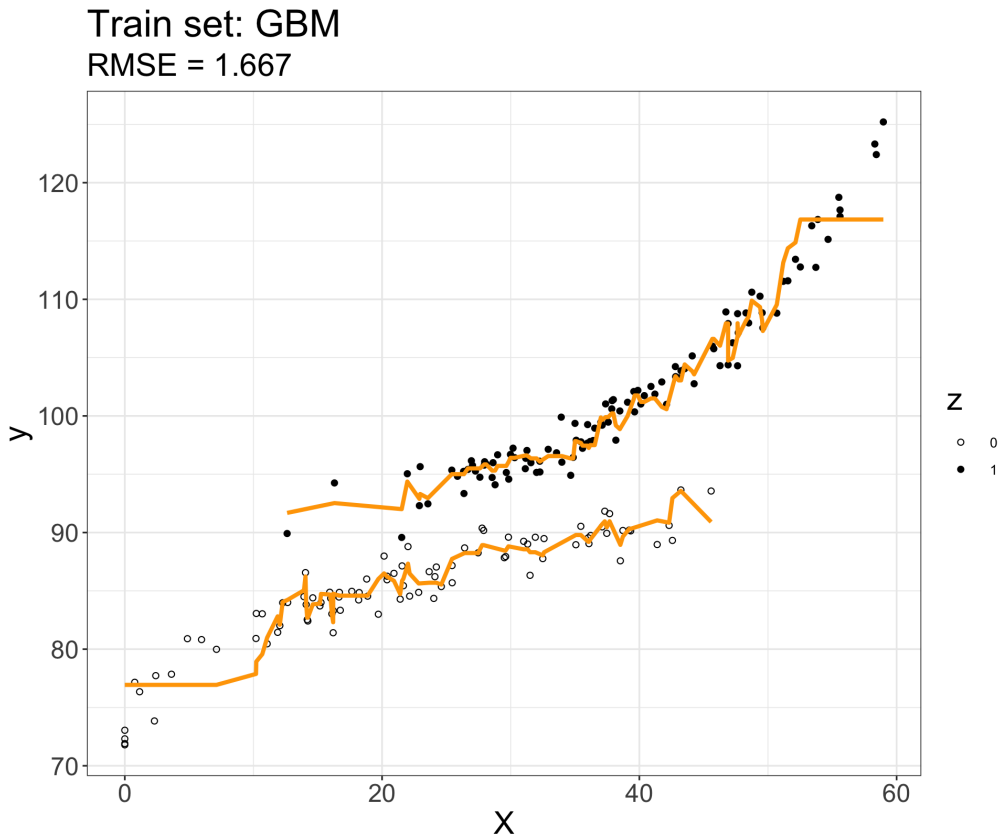
Gradient Boosted Regression Trees: Tree 50

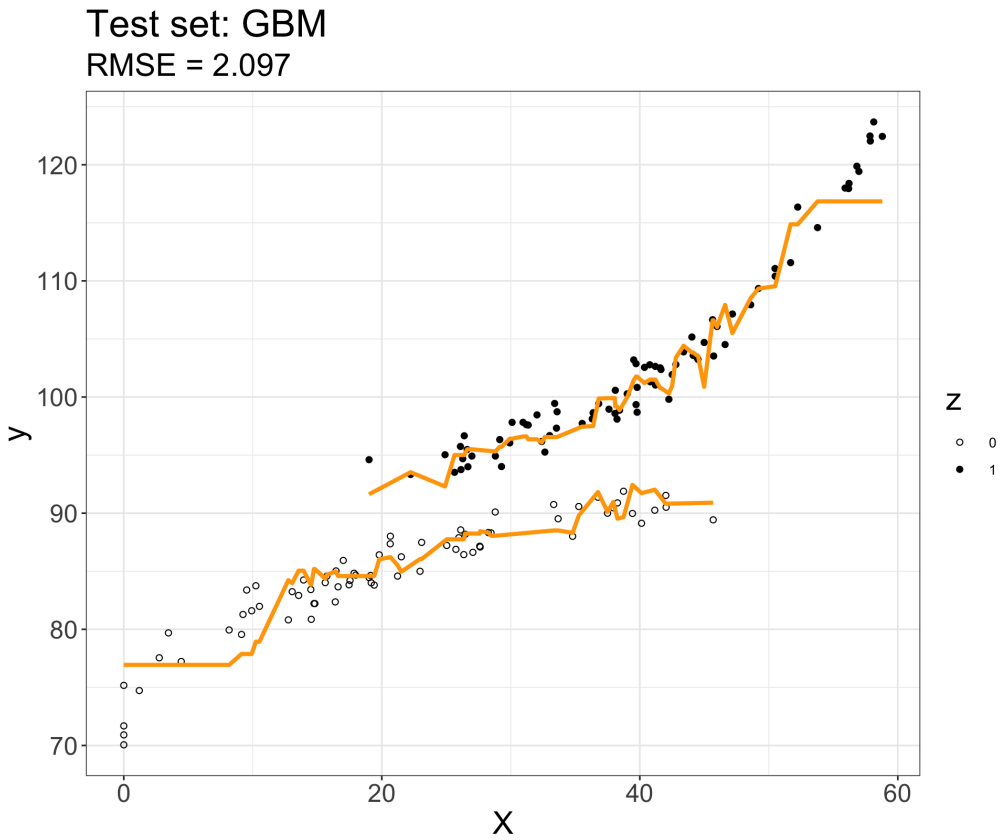


Gradient Boosted Regression Trees Result



Gradient Boosted Regression Trees Result





BART

Bayesian Additive Regression Trees a flexible non-parametric machine learning model with coherent uncertainty intervals.

$$Y_i = f(\mathbf{x}_i, z) + \epsilon_i, \quad \epsilon_i \overset{\text{iid}}{\sim} N(0, \sigma^2)$$

$$f(\mathbf{x}, z) = g(\mathbf{x}, z, T_1, M_1) + g(\mathbf{x}, z, T_2, M_2) + \cdots + g(\mathbf{x}, z, T_m, M_m)$$

g represents regularization from the BART prior

T represents a given regression tree

M represents means in terminal nodes (leafs) for tree m

MCMC draws of a posterior distribution from a Gibbs Sampler

Chipman, George & McCulloch 2007, 2010

Bayesian Boosting the Logic

Trees are proposed and either accepted or rejected with at Metropolis-Hastings Algorithm.

With Bayesian Boosting, we trade traditional hyper-parameters like tree depth or learning rate for Bayesian priors.

What are our prior beliefs about regression trees?

Chipman, George & McCulloch 2007, 2010

Bayesian Boosting the Logic

Trees are proposed and either accepted or rejected with at Metropolis-Hastings Algorithm.

With Bayesian Boosting, we trade traditional hyper-parameters like tree depth or learning rate for Bayesian priors.

What are our prior beliefs about regression trees?

- We should be skeptical about predictions really far from the average

Chipman, George & McCulloch 2007, 2010

Bayesian Boosting the Logic

Trees are proposed and either accepted or rejected with at Metropolis-Hastings Algorithm.

With Bayesian Boosting, we trade traditional hyper-parameters like tree depth or learning rate for Bayesian priors.

What are our prior beliefs about regression trees?

- We should be skeptical about predictions really far from the average
- Trees should be small (bigger trees are more likely to overfit)

Chipman, George & McCulloch 2007, 2010

Bayesian Boosting the Logic

Trees are proposed and either accepted or rejected with at Metropolis-Hastings Algorithm.

With Bayesian Boosting, we trade traditional hyper-parameters like tree depth or learning rate for Bayesian priors.

What are our prior beliefs about regression trees?

- We should be skeptical about predictions really far from the average
- Trees should be small (bigger trees are more likely to overfit)
- We believe our model should be able to outperform a linear regression

Chipman, George & McCulloch 2007, 2010

Priors

3 independent priors on μ (means) T_j (trees) and σ (error)

These priors are calibrated from the data and keep the model from overfitting.

The priors give BART strong predictive performance without cross validation.

Prior on μ

When running a BART model the outcome is transformed to be within -.5 and .5.

For each terminal node μ, let $\mu \sim N(0, \sigma_\mu^2)$

For any individual $\mathbb{E}(y|x)$ is just the sum of the fit across all terminal nodes (leafs) for m trees (assuming independence). This makes $N(0, \sigma_\mu^2)$ and appropriate prior for $\mathbb{E}(y|x)$.

The idea is to set σ_μ^2 so that ~ 95% of all the values fall within (-.5, .5).

We can calibrate this prior by making the following true.

$$0 + k\sqrt(m)\sigma_\mu = .5 = \frac{.5}{k\sqrt(m)}$$

$k = 2$ will give us about a 95% probability of values within -.5 and .5

Generally k is the number of standard deviations of $\mathbb{E}(y|x)$ from a mean of 0 to the boundary at .5.

Prior on T

Another way to over fit is by allowing trees to grow to deep. In boosted regression trees max tree depth is set or cross validated. In BART, it is learned from the data and regularized with a prior!

$\alpha(1+d)^{-\beta}, \alpha \in (0,1), \beta \in [0,\infty)$

Default settings of $\alpha = .95$ and $\beta = 2$ give the following:

depth of 1 = .05

depth of 2 = .55

depth of 3 = .28

depth of 4 = .09

depth of 5 = .03

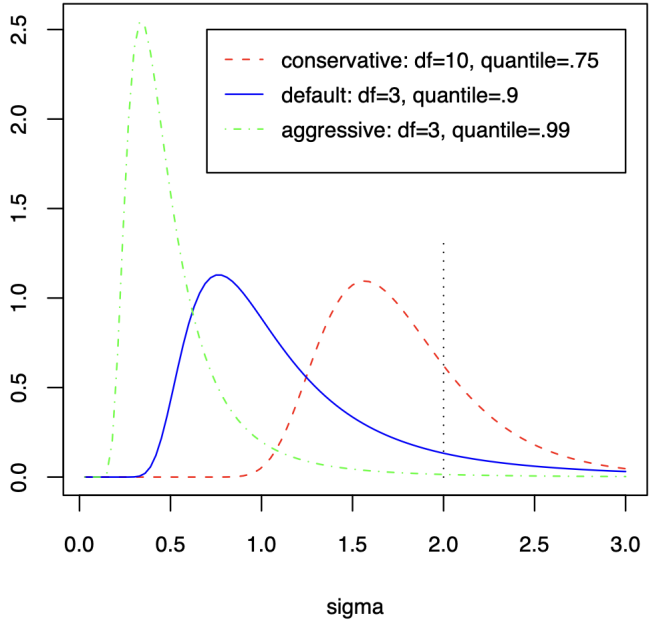
Trees will grow deeper if the data demands it!!!

Prior on σ

The goal is to calibrate expectations about the residual standard deviation relative to a reasonable but conservative benchmark $\hat{\sigma}_\epsilon$.

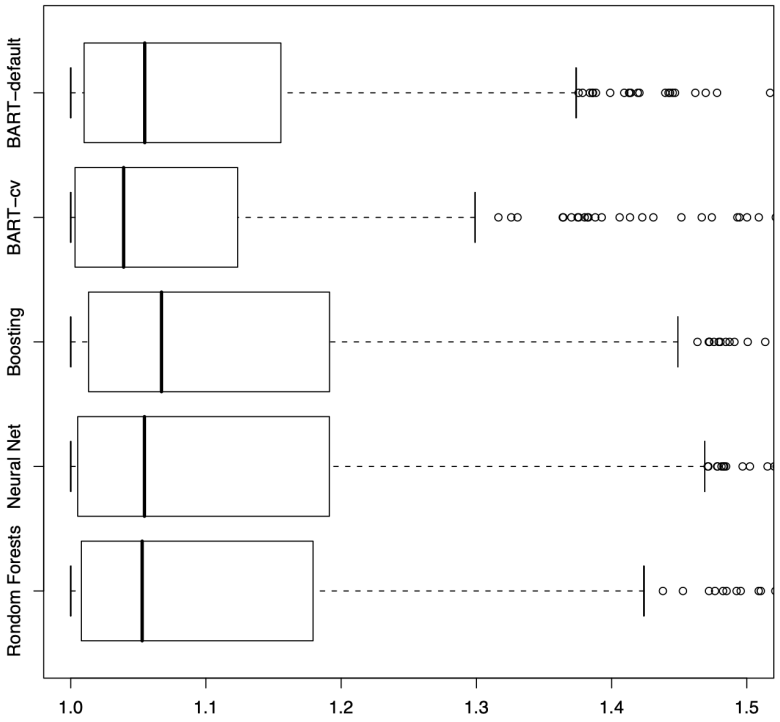
Then choose df for a χ^2 distribution and choose the quantile you'd like $\hat{\sigma}_\epsilon$ to represent.

Least squares is used for the initial rough guess of $\hat{\sigma}_\epsilon$. This is the dotted black line. With the default setting the prior assumes a 90% probability of σ being smaller than OLS regression.



BART and Prediction Performance

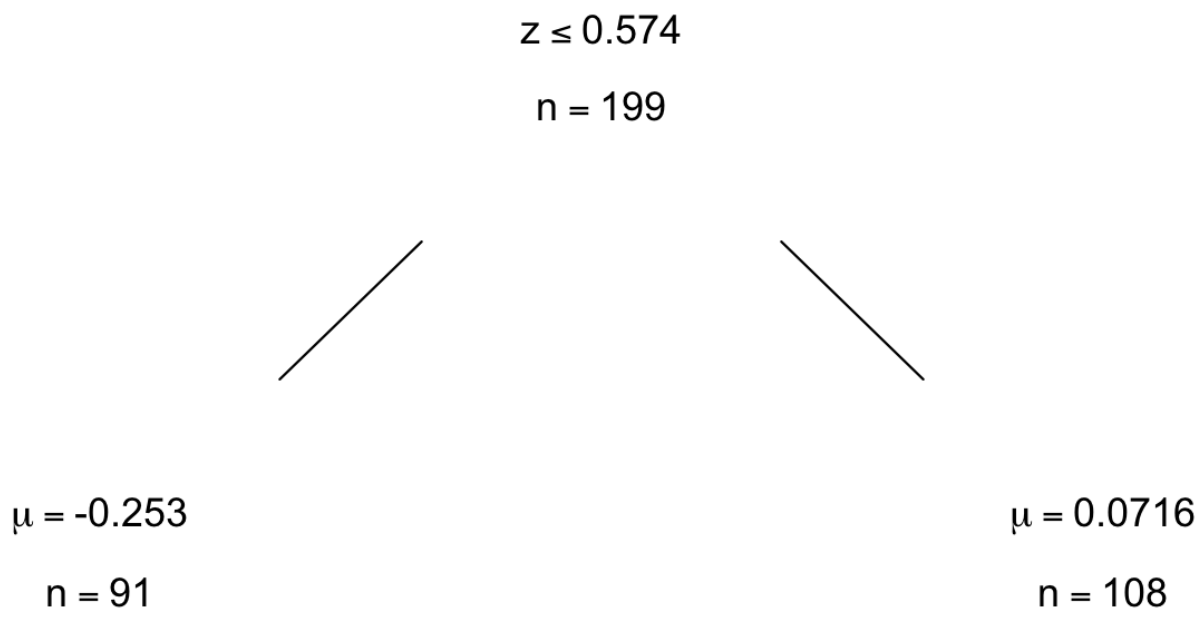
Across 42 data sets and over 800 train test splits BART without cross validation performs better than the competitors. This performance increases when BART priors are cross valdiated. (Chipman, George, McCulloch 2007, 2010)



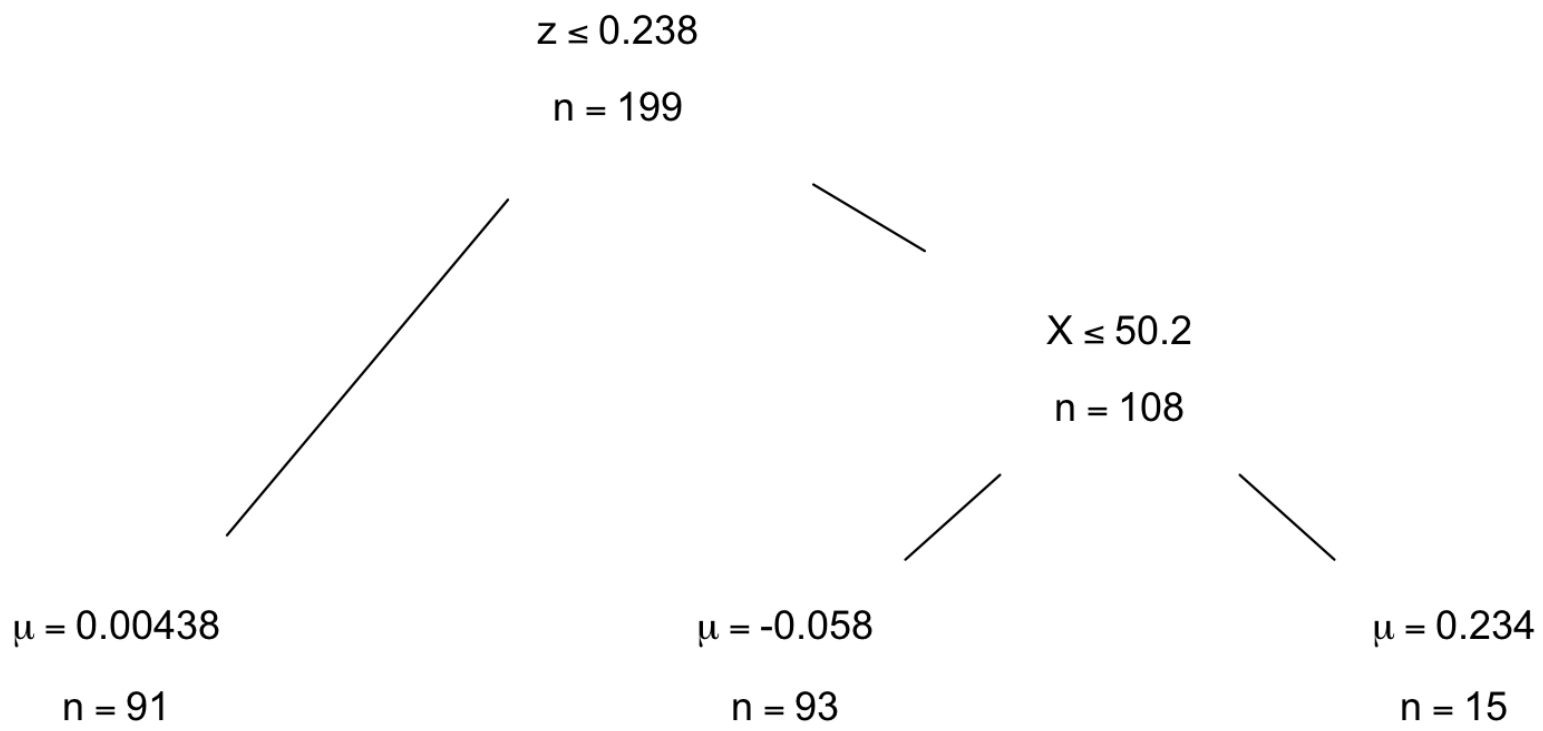
Fitting a BART Model

1. Start with the scaled mean of the outcome
2. Propose a variety of trees (grow a new tree, prune a tree, flip a tree, add a split)
3. Accept or reject trees
4. Repeat over many (thosands) of iterations

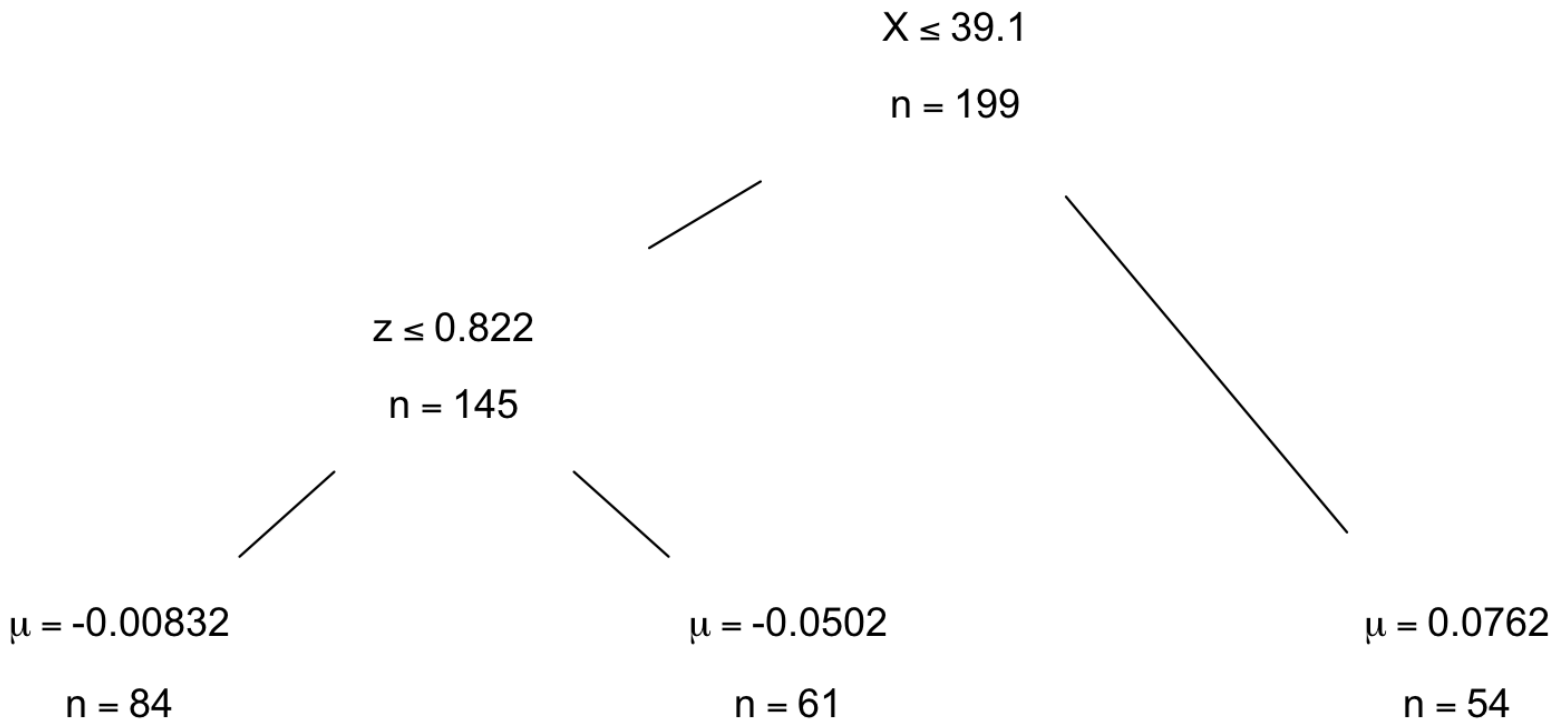
Updating Trees: an accepted tree



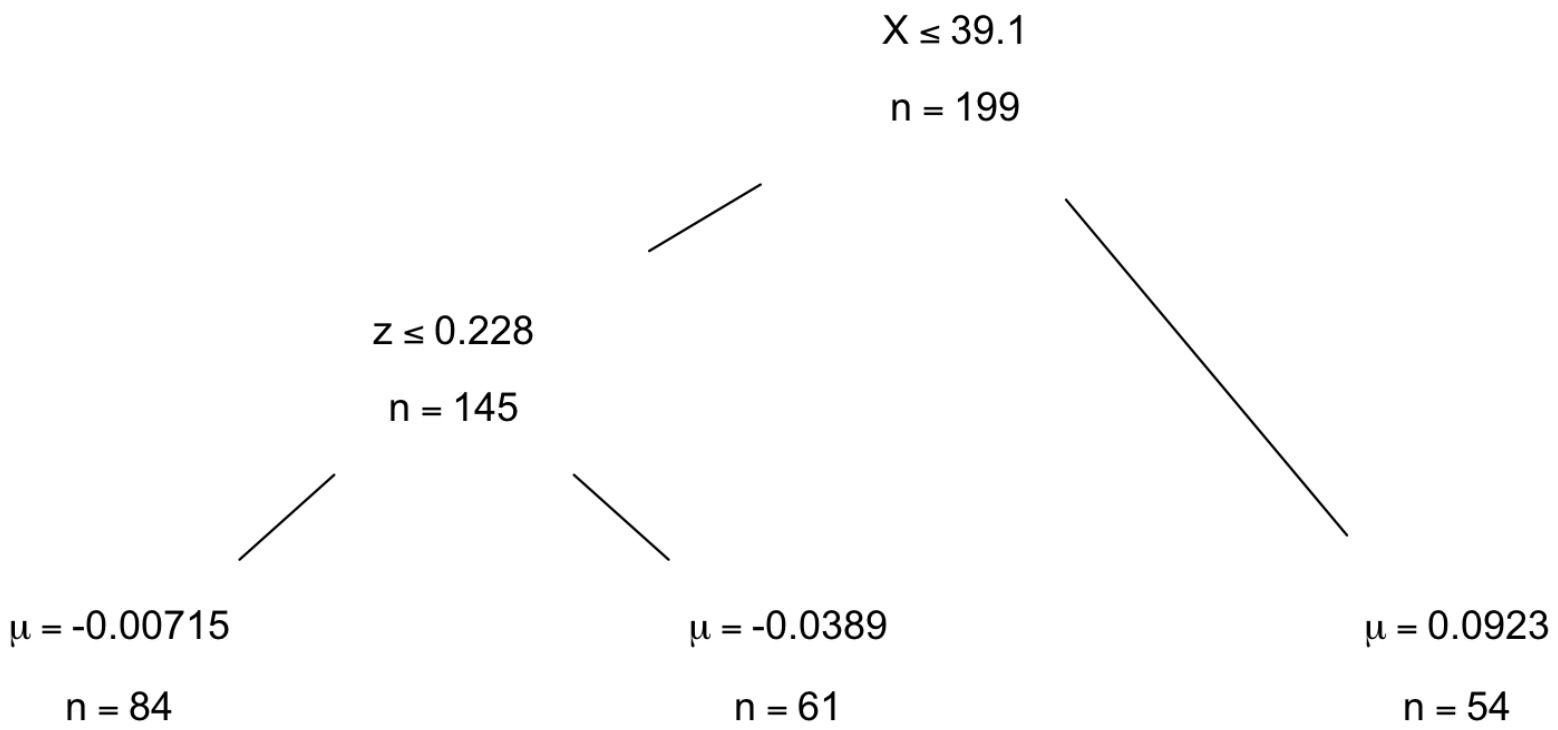
Updating Trees: proposal to add split on X is accepted!



Updating Trees: proposal to flip tree is accepted!



Updating Trees: new proposal rejected!

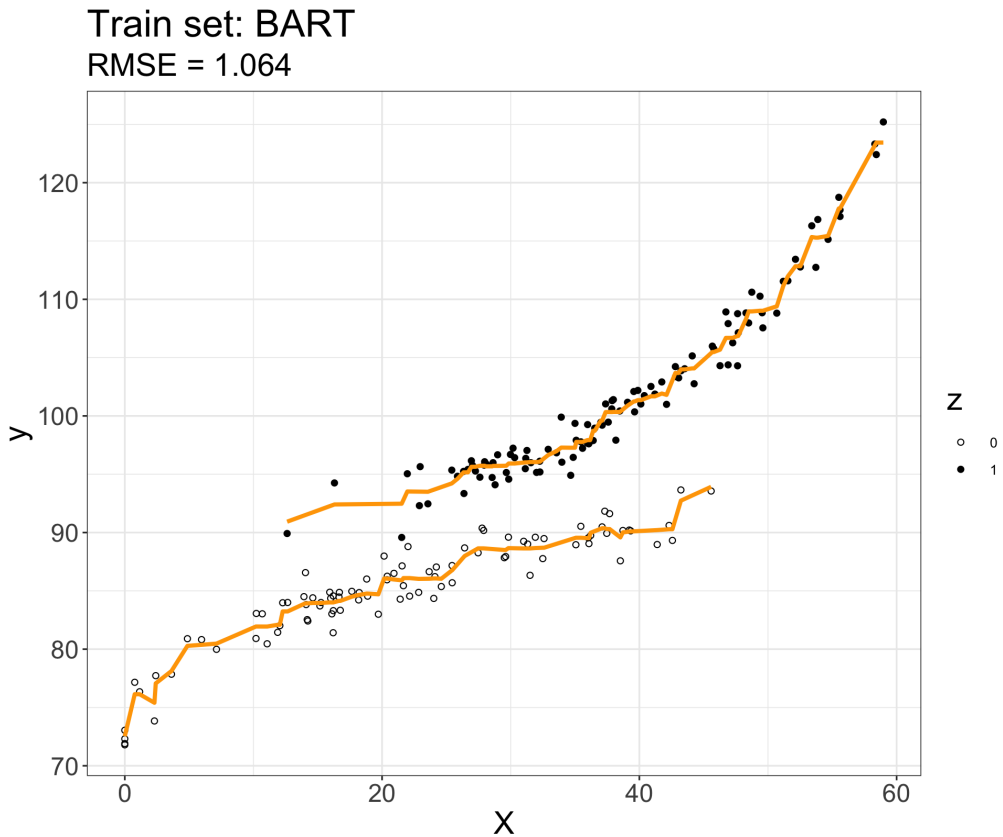


Updating Trees: new proposal to prune is accepted!

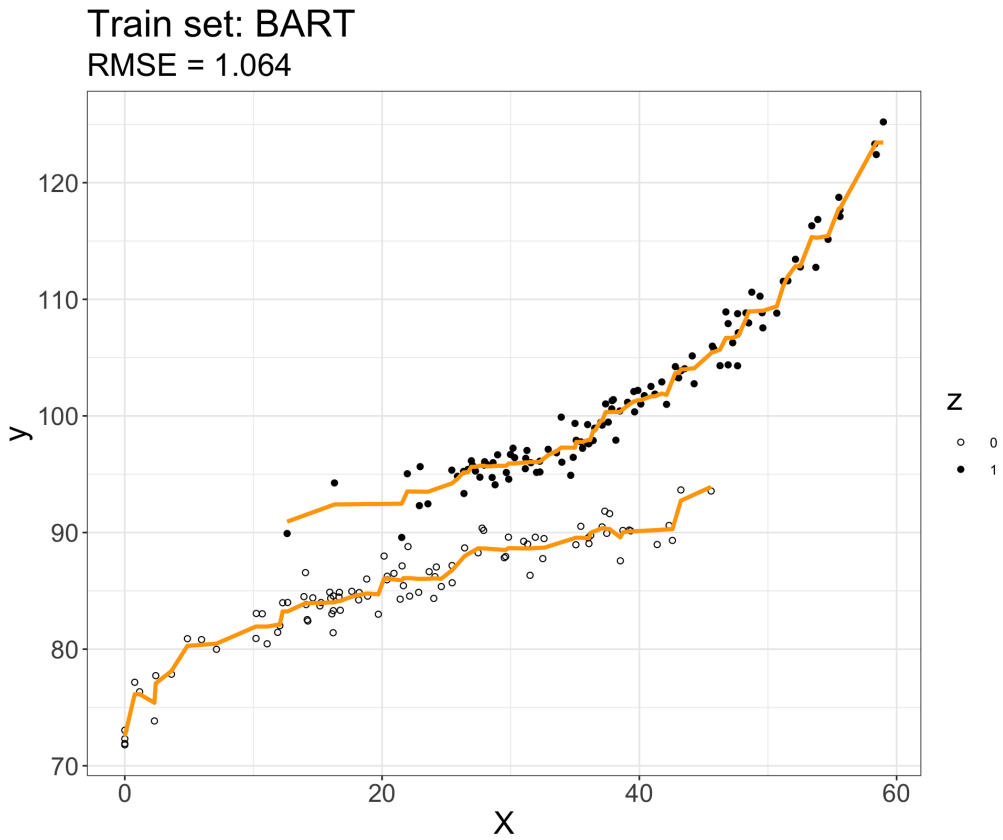
$\mu = -0.00461$

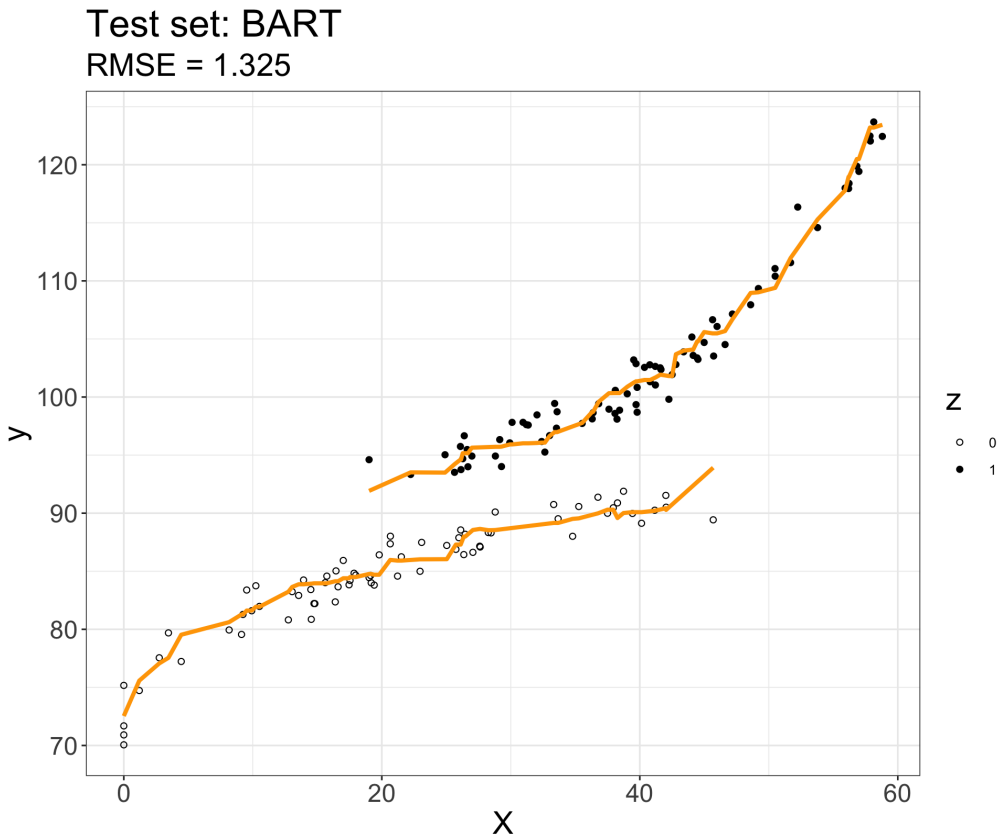
$n = 199$

Bayesian Additive Regression Trees Results

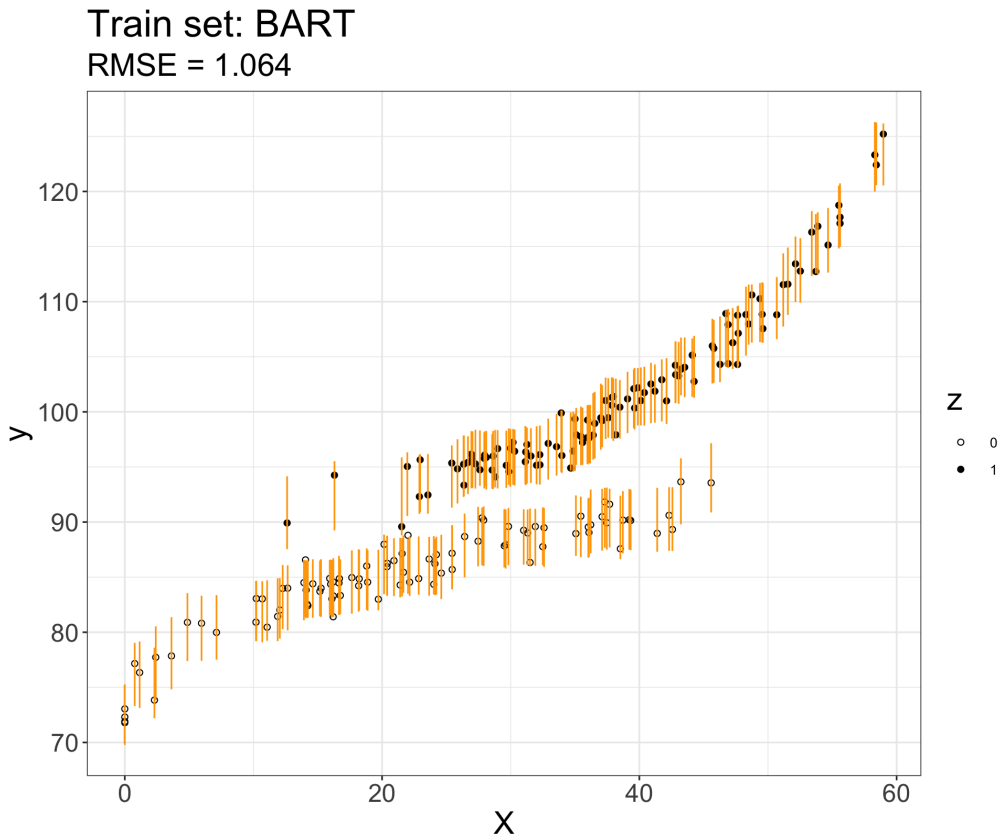


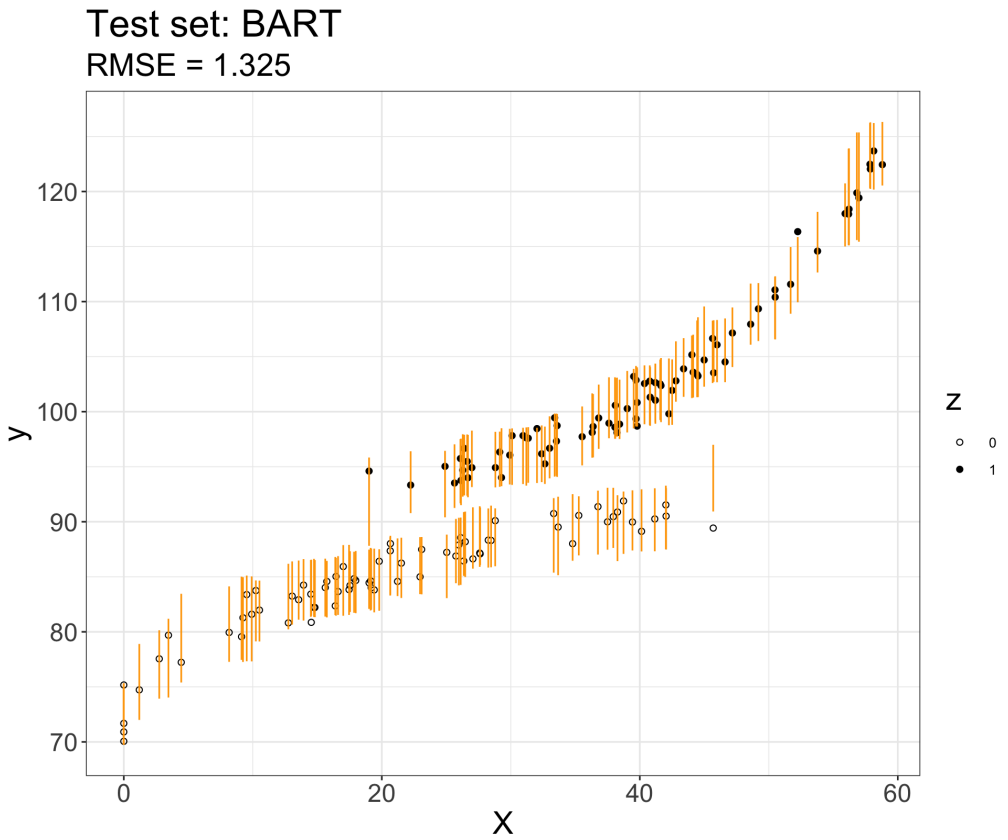
Bayesian Additive Regression Trees Results



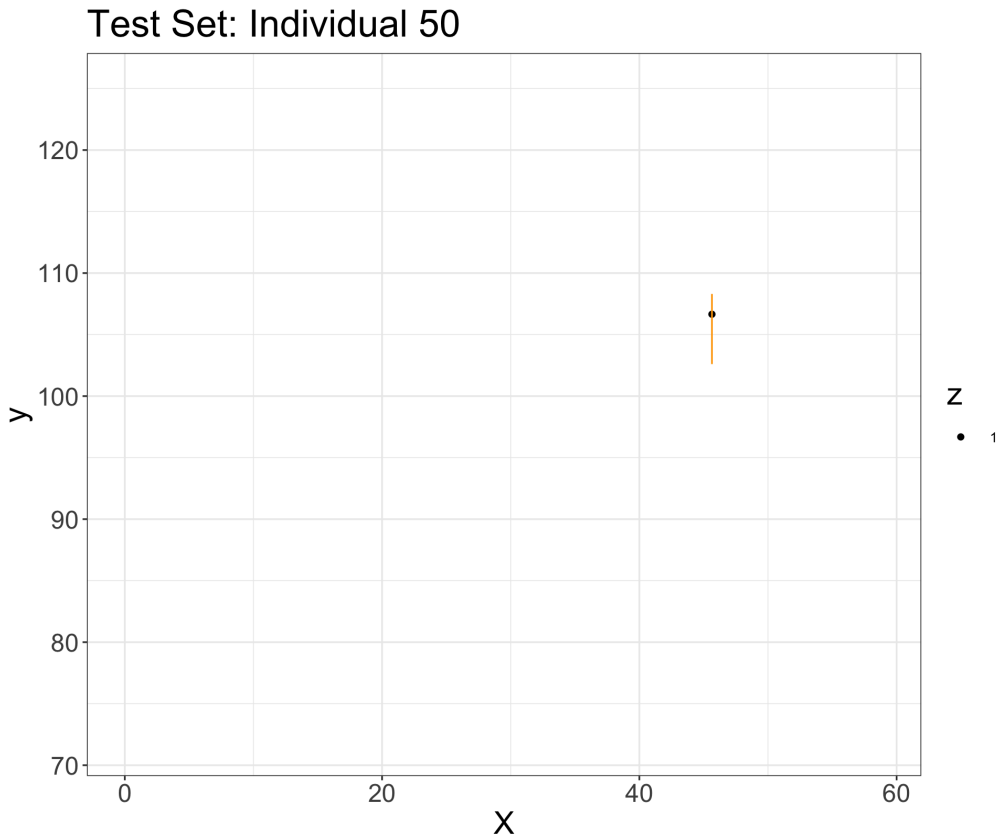


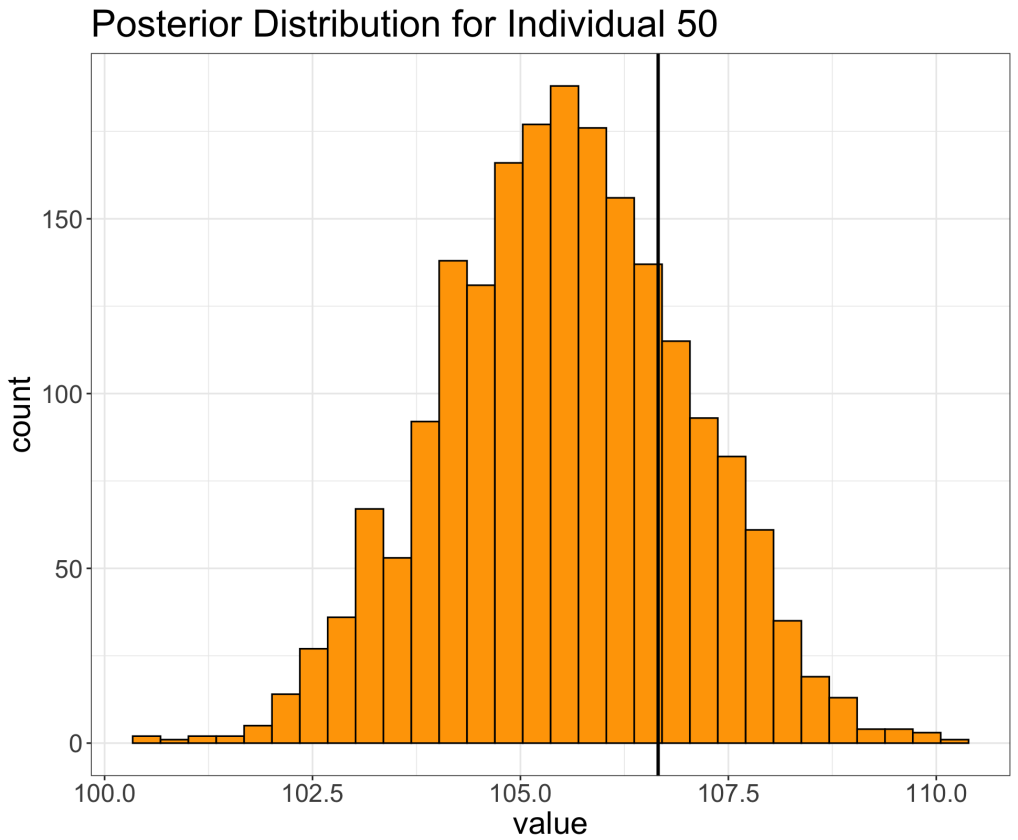
BART Predictions are Posterior Distributions



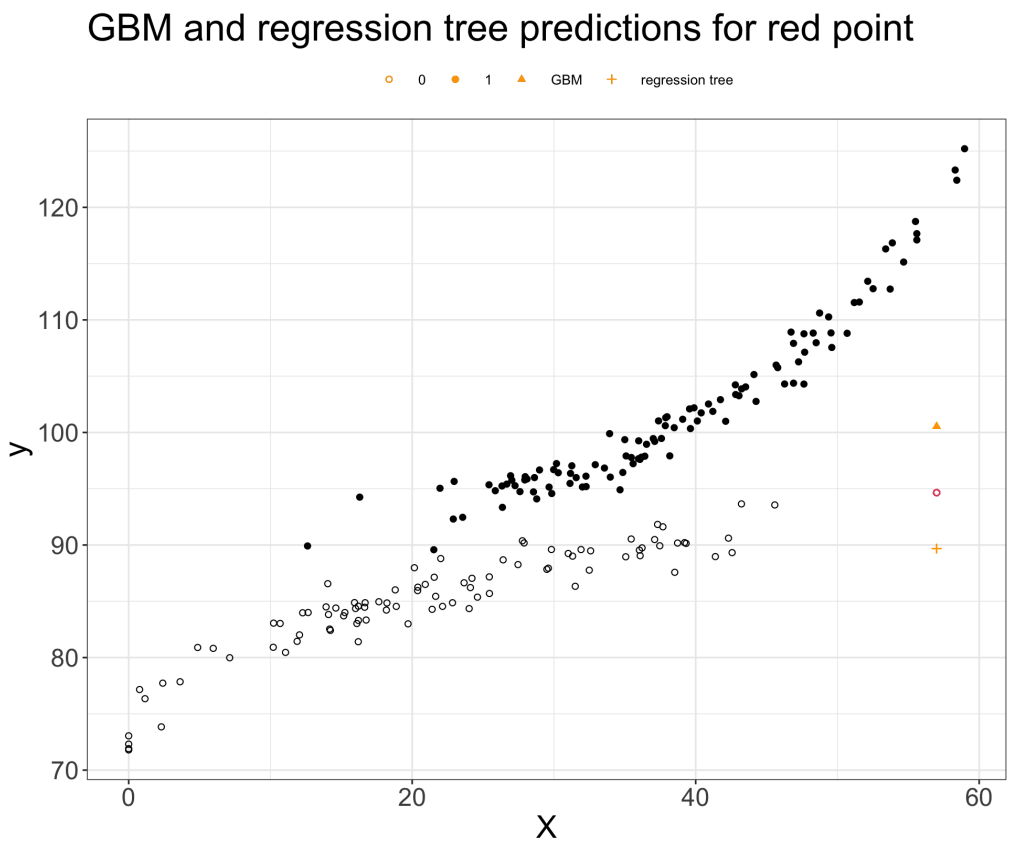


Zooming in on a Single Case

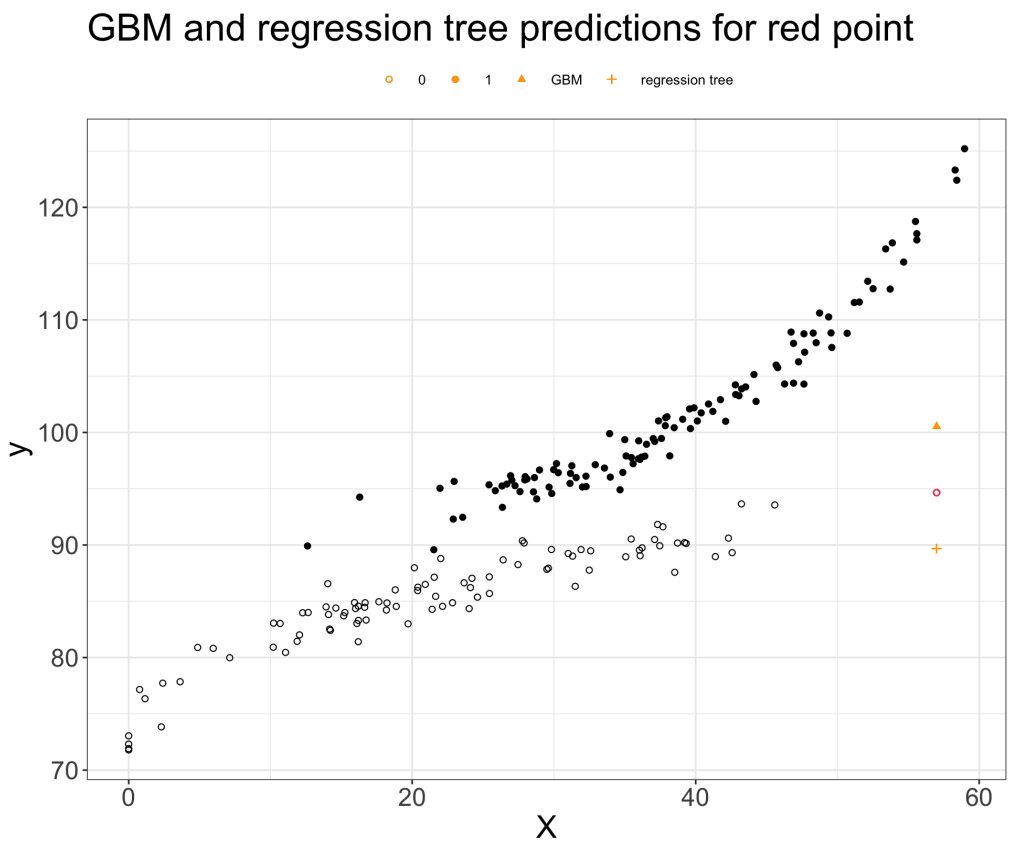


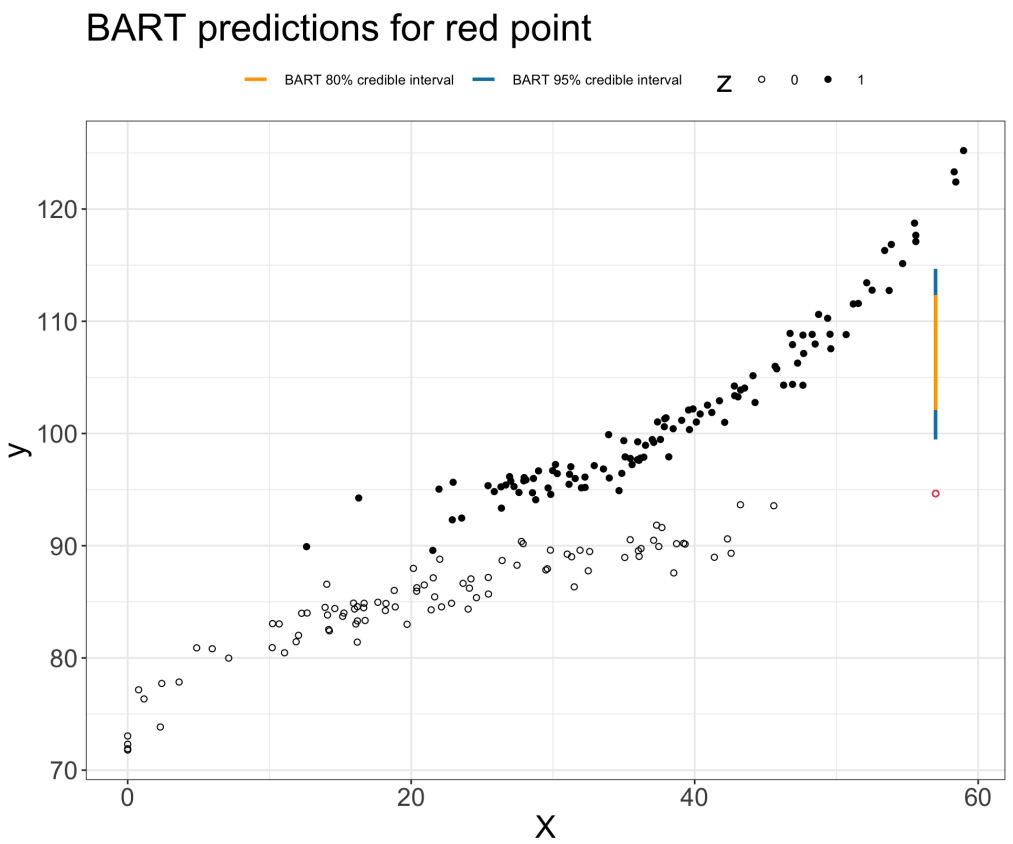


Predictions Beyond the Range of our Training Data



Predictions Beyond the Range of our Training Data





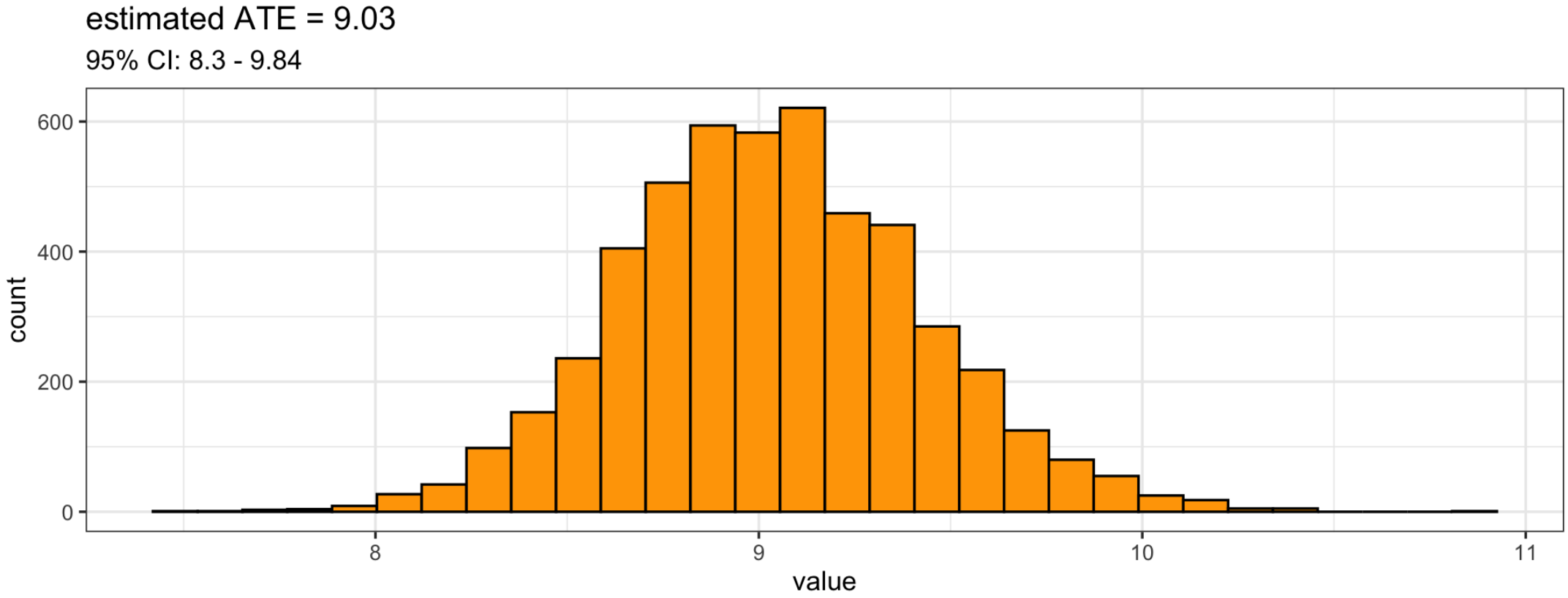
BART for Causal Inference

What if we were not interested in making predictions but in the causal effect of Z on Y?

BART for Causal Inference

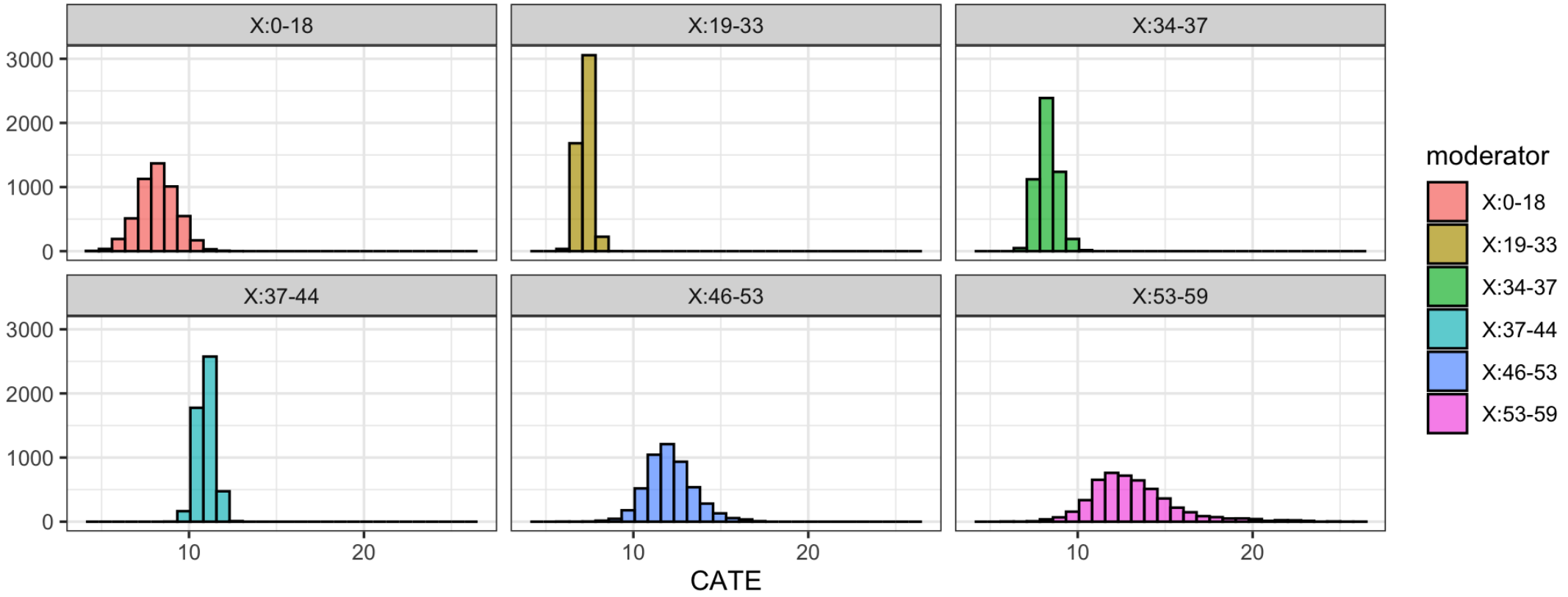
What if we were not interested in making predictions but in the causal effect of Z on Y?

Causal inference is a prediction problem! If the appropriate causal assumptions are met, we can estimate the average causal effect (or any other average treatment effect).



Treatment Effect Moderation

BART will automatically detect treatment effect moderators!



Bayesian Boosting with R!

```
library(dbarts) # for prediction (bart2() function)

library(bartCause) # for causal inference

library(stan4bart) # multilevel models and BART

library(thinkCausal) # point and click causal inference with BART

remotes::install_github('priism-center/thinkCausal_dev', subdir = 'thinkCausal')
```