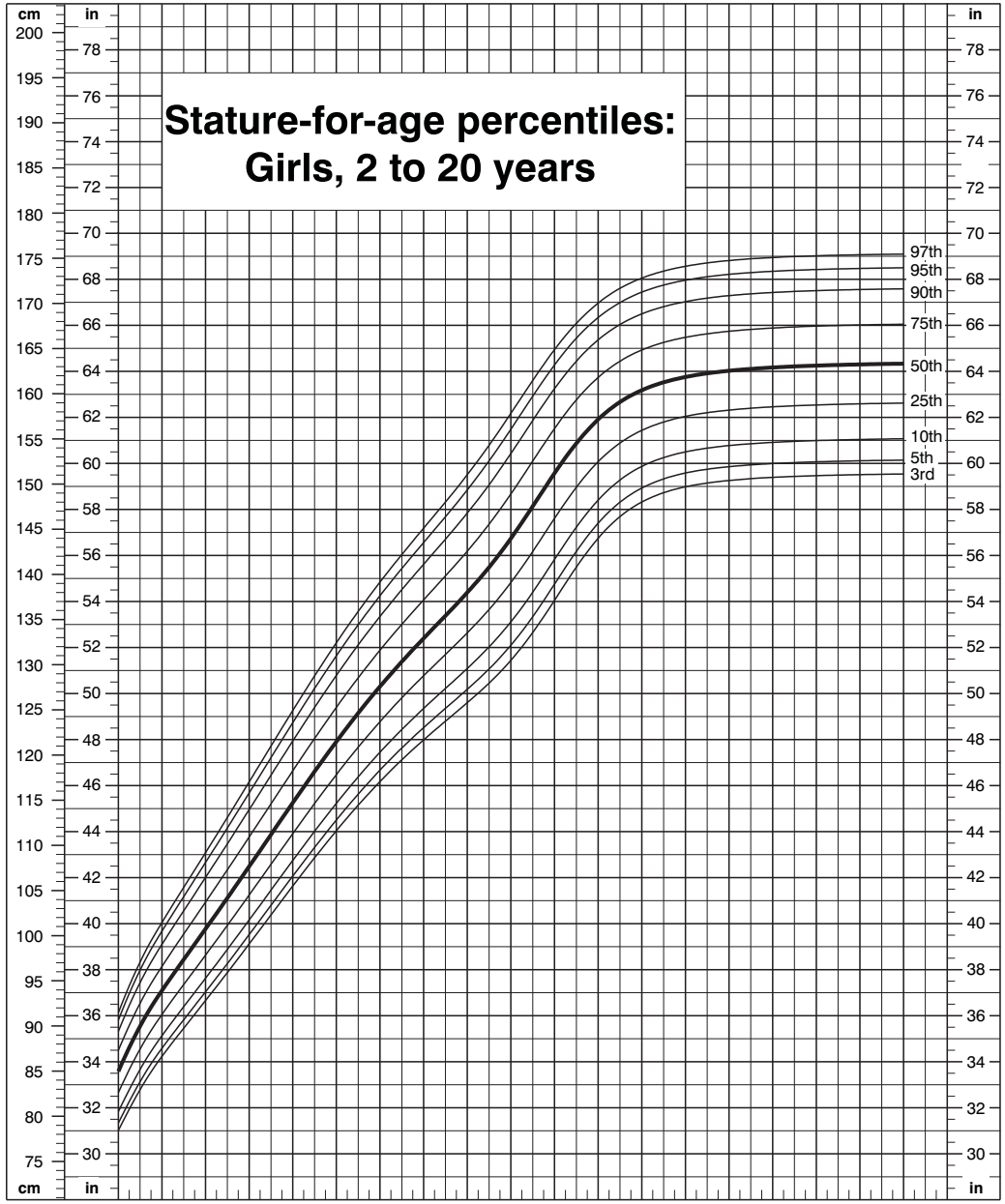
An introduction to Bayesian Additive Regression Trees (BART)

Rodney Sparapani
Associate Professor of Biostatistics
Medical College of Wisconsin

September 15, 2025

CDC Growth Charts: United States



Motivating Example: Growth Charts

- The US Centers for Disease Control and Prevention (CDC) as well as the World Health Organization have developed growth charts for childhood development: height by age, weight by age, body mass index by age and weight by height
- Here we will focus on height, y_t, by age in months, $t = 24, \ldots, 215$ (2 to 17 years old)
- The CDC uses the LMS method via natural cubic splines (Cole and Green 1992 *Statistics in Medicine*)
- Three parameters estimated by penalized maximum likelihood the Box-Cox power transformation, L_t; the mean, M_t; and the coefficient of variation, S_t

$$z_t = \left\{ \begin{array}{ll} \frac{-1+(y_t/M_t)^{L_t}}{L_t S_t} & L_t \neq 0 \\ \frac{\log(y_t/M_t)}{S_t} & L_t = 0 \end{array} \right\} \sim \mathrm{N}(0, 1)$$

- But, this only uses part of the data: just males or just females
- Male/female trajectories are quite similar until about age 12
- So what if we wanted to use all of the data?

What is Machine Learning Regression?

- The more appropriate term is "Statistical Learning" but "Machine Learning" has caught on so we are stuck with it
- Machine Learning Regression (MLR) is within the paradigm of Artificial (or Computational) Intelligence
- MLR is extensible, but for the moment consider the general regression case of a continuous outcome with Normal errors

$$y_i = \mu + f(x_i) + \epsilon_i \qquad \text{where } \epsilon_i \overset{\text{iid}}{\sim} \mathrm{N}\left(0,\ \sigma^2\right)$$

- f is an unspecified function whose form is to be *learned* from the data and $\boldsymbol{x_i}$ is a vector of covariates for $i = 1, \ldots, N$
- *Ideally*, in a *nonparametric* manner without resorting to *precarious restrictive assumptions*

What is Bayesian Additive Regression Trees?

- a supervised MLR with nice properties: automated learning of the functional relationship and interactions without requiring covariate transformations for continuous, binary, categorical and time-to-event outcomes
- tree-based ensemble predictive model
- Bayesian nonparametric method with robust defaults for the prior parameter settings
- computationally efficient posterior inference via MCMC estimates naturally computed from summaries of the posterior along with the quantification of their uncertainty
- seamless extension to variable selection in high dimensions

Selected BART references with URLs

Overview	Chipman, George and McCulloch 2010 *AOAS* Sparapani, Spanbauer and McCulloch 2021 *JSS*
Survival Analysis	Sparapani, Logan et al. 2016 *Statistics in Medicine* Henderson, Louis et al. 2020 *Biostatistics* Sparapani, Rein et al. 2020 *Biostatistics* Sparapani, Logan et al. 2020 *SMMR* Linero, Basak et al. 2021 *Bayesian Analysis*
Big Data (Big N)	Pratola, Chipman et al. 2014 *JCGS* Entezari, Craiu et al. 2017 *Canadian J of Stat*
Variable Selection (Big P)	Linero 2018 *JASA* Liu, Rockova 2021 *JASA*
Efficient MCMC	Pratola 2016 *Bayesian Analysis*
Nonparametric Theory	Rockova and Saha 2019 *PMLR* Rockova and van der Pas 2020 *AOS*
Heteroskedastic	Pratola, Chipman et al. 2020 *JCGS*
Propensity Scores	Hahn, Murray et al. 2020 *Bayesian Analysis*
Monotonic	Chipman, George et al. 2021 *Bayesian Analysis*

Bayesian Additive Regression Trees (BART)

Chipman, George & McCulloch 2010 *Annals of Applied Stat*

$$y_i = f(x_i) + \epsilon_i \qquad \epsilon_i \overset{\text{iid}}{\sim} \mathrm{N}\left(0,\ w_i^2\sigma^2\right)$$

$$f \overset{\text{prior}}{\sim} \mathbf{BART}\ (H, \mu, \kappa, \tau, \alpha, \beta)$$

$$f(x_i) \equiv \mu + \sum_{h=1}^{H} g(x_i; \mathcal{T}_h, \mathcal{M}_h) \qquad H \in \{50, 200, 500\}$$

$$\mu_{hl} | \mathcal{T}_h \overset{\text{prior}}{\sim} \mathrm{N}\left(0,\ \frac{\tau^2}{4H\kappa^2}\right) \text{ leaves of } \mathcal{T}_h$$

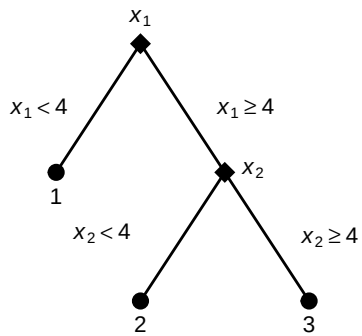
$$\in \mathcal{M}_h$$

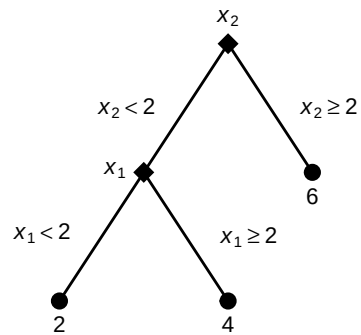
$$\sigma^2 \overset{\text{prior}}{\sim} \lambda\nu\chi^{-2}(\nu)$$

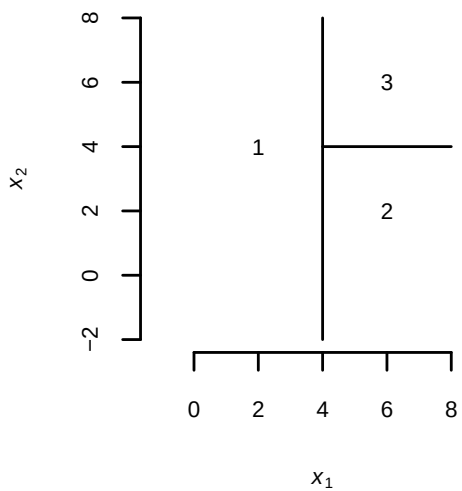
$$\overset{\text{prior}}{\sim} \mathbf{Gamma}^{-1}(\nu/2,\ \lambda\nu/2) \qquad \mathrm{E}\left[\sigma^2\right] = \lambda\nu/(\nu - 2)$$

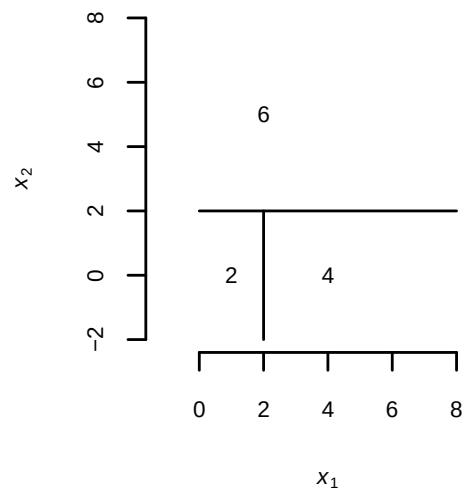
Bayesian Additive Regression Trees (BART)

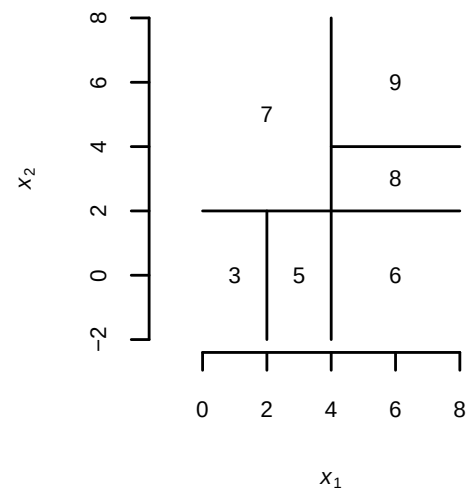
Logan, Sparapani, McCulloch & Laud 2020 *SMMR*











BART, ensembles and prediction error

- mean squared error = bias^2 + variance
- There is a trade-off between the bias and variance
- Consider the spectrum of trade-offs

  Linear regression is on the high bias/low variance end

  Single-tree regression is on the low bias/high variance end
- Ensembles are in the middle: medium bias/medium variance
- BART is in the class of ensemble models which both theoretically, and in practice, have excellent out-of-sample predictive performance

Krogh & Solich 1997 *Physical Review E*
Baldi & Brunak 2001 "Bioinformatics: machine learning approach"
Kuhn & Johnson 2013 "Applied Predictive Modeling"

Binary trees and Bayesian Additive Regression Trees

- BART relies on an ensemble of $\boldsymbol{H}$ binary trees
- We exploit the wooden tree metaphor to its fullest except binary trees grow downward by tradition
- Each of these trees grows down starting out as a root node
- The root node is generally a branch decision rule, but it doesn't have to be; occasionally, there are trees in the ensemble which are only a root terminal node consisting of a single leaf output value
- If the root is a branch decision rule, then it spawns a left and a right node which each can be either a branch decision rule or a terminal leaf value and so on
- In binary tree, $\mathcal{T}$, there are $\boldsymbol{C}$ nodes which are made of $\boldsymbol{B}$ branches and $\boldsymbol{L}$ leaves: $\boldsymbol{C = B + L}$
- There is an algebraic relationship between the number of branches and leaves which we express as $\boldsymbol{B = L - 1}$.

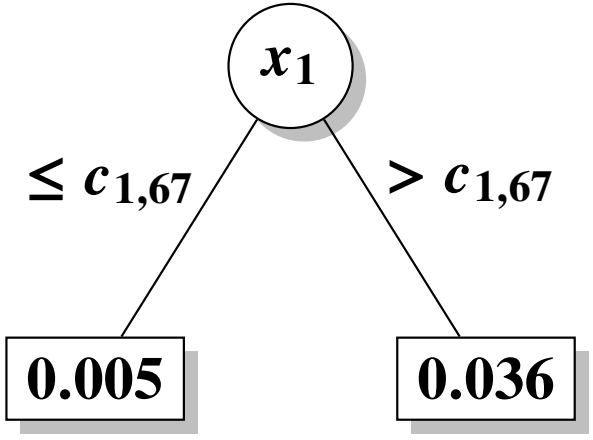
The BART R package

- to facilitate the `predict` function, BART fits can be stored as R objects to be reloaded later
- the ensemble of trees is encoded in an ASCII string which is returned in the `treedraws$trees` list item
- This string can be read by R
- Encoded with C/C++ indexing starting with 0 is used rather than R object indexing starting with 1
- Since the `predict` function calls C/C++ code for speed

The BART R package and trees

Sparapani, Spanbauer and McCulloch 2021
Journal of Statistical Software

```
R> write(post$treedraws$trees, "trees.txt")
R> tc <- textConnection(post$treedraws$tree)
R> trees <- read.table(file=tc, fill=TRUE, row.names=NULL,
+    col.names=c("node", "var", "cut", "leaf"))
R> close(tc)
R> head(trees)
  node var cut          leaf
1 1000 200   1            NA
2    3  NA  NA            NA
3    1   0  66 -0.001032108
4    2   0   0  0.004806880
5    3   0   0  0.035709372
6    3  NA  NA            NA
```



The BART R package and trees

- The `treedraws$trees` string is encoded as follows
- The first line is an exception which has the number of MCMC samples, M, in the field `node`; the number of trees, H, in the field `var`; and the number of variables, P, in the field `cut`
- For the rest of the file, the field `node` is used for the number of nodes in the tree when all other fields are `NA`; or for a specific node when the other fields are present
- The nodes are numbered in relation to the tree's depth level, $d(n) = \lfloor \log_2 n \rfloor$ or `floor(log2(node))`

Depth							
0				1			
1		2				3	
2	4		5		6		7
⋮							
d		2^d		…		$2^{d+1}-1$	

The BART R package and trees

- The `var` field is the variable in the branch decision rule which is encoded $\mathbf{0}, \ldots, \boldsymbol{P-1}$ as a C/C++ index (rather than R)
- Similarly, the `cut` field is the cut-point of the variable in the branch decision rule which is encoded $\mathbf{0}, \ldots, \boldsymbol{c_j - 1}$ for variable $\boldsymbol{j}$
- cut-points are returned in the `treedraws$cutpoints` list item
- The terminal leaf output value is contained in the field `leaf`
- It is not immediately obvious which nodes are branches vs. leaves since, at first, it would appear that the `leaf` field is given for both branches and leaves
- Confusingly: leaves are always associated with `var=cut=0`
- however, note that this is also a valid branch variable/cut-point since these are C/C++ indices

The BART R package and trees

- The key to proper discrimination between branches and leaves is via the algebraic relationship between a branch, $\boldsymbol{n}$, at tree depth $\boldsymbol{d(n)}$ leading to its left, $\boldsymbol{l = 2n}$, and right, $\boldsymbol{r = 2n + 1}$, nodes at depth $\boldsymbol{d(n) + 1}$
- for each node, besides root, you can determine from which branch it arose and those nodes that are not a branch (since they have no leaves) are necessarily leaves

The BART prior

- The BART prior specifies a flexible class of unknown functions, f, from which we can gather randomly generated fits to the given data via the posterior
- Here we define f as returning a scalar value: for a multivariate BART example, see Um, Linero, et al. 2023 *Stat in Med*
- Let function $g(x; \mathcal{T}, \mathcal{M})$ assign a value based on the input x
- The binary tree $\mathcal{T}$ is represented by a collection of C four-tuples $(n, \psi_n; j, k)$: n for node number;
  $\psi_n = 1$ for a branch and 0 for a leaf;
  and, if a leaf, j for covariate x_j with k for the cut-point c_{jk}
- The collection of branches is denoted by $\mathcal{B} = \{n : \psi_n = 1\}$
- The branch decision rules are of the form $x_j \leq c_{jk}$ which means branch left and $x_j > c_{jk}$, branch right; or terminal leaves where it stops.
- $\mathcal{M}$ represents leaves and is a set of ordered pairs, (n, μ_n):
  $n \in \mathcal{L}$ where $\mathcal{L}$ is the set of leaves
  ($\mathcal{L}$ is the complement of $\mathcal{B}$) and μ_n for the outcome value

The BART prior

- The function, $f(x)$, is the following sum:

$$f(x) = \mu + \sum_{h=1}^{H} g(x; \mathcal{T}_h, \mathcal{M}_h)$$

  where H is "large", let's say, 50, **200** or 500
- For a continuous outcome, y_i, we have the following BART regression on the vector of covariates, x_i:

$$y_i = f(x_i) + \epsilon_i \text{ where } \epsilon_i \overset{\text{iid}}{\sim} \mathrm{N}\left(0,\ w_i^2 \sigma^2\right)$$

  with i indexing subjects $i = 1, \ldots, N$
- The BART priors for the unknown random function, f, and the error variance, σ^2, are as follows

$$f \overset{\text{prior}}{\sim} \text{BART}\ (H, \mu, \kappa, \tau, \alpha, \beta) \qquad \sigma^2 \overset{\text{prior}}{\sim} \nu\lambda\chi^{-2}\ (\nu)$$

  where H is the number of trees, μ is a known constant which centers y and the rest of the parameters will be explained later in this section (for brevity, we often use $f \overset{\text{prior}}{\sim} \text{BART}$)

The BART prior

- The w_i are known standard deviation weight multiples which you can supply with the argument `w` that is only available for continuous outcomes, hence, the weighted BART name; the unit weight vector is the default
- The centering parameter, μ, can be specified via the `fmean` argument where the default is taken to be $\bar{y}$
- x_i: matrices or data frames can be supplied
- unlike matrices, data frames can contain categorical factors when `x.train` is a data frame
- Factors with multiple levels are transformed into dummy variables with each level as their own binary indicator; factors with only two levels are a binary indicator with a single dummy variable

The BART prior

- BART is a Bayesian nonparametric prior
- we represent the BART prior in terms of the collection of all trees, $\mathcal{T}$; collection of all leaves, $\mathcal{M}$; and the error variance, σ^2, as: $[\mathcal{T}, \mathcal{M}, \sigma^2] = [\sigma^2]\,[\mathcal{T}, \mathcal{M}] = [\sigma^2]\,[\mathcal{T}]\,[\mathcal{M}|\mathcal{T}]$
- the individual trees themselves are independent: $[\mathcal{T}, \mathcal{M}] = \prod_h [\mathcal{T}_h]\,[\mathcal{M}_h|\mathcal{T}_h]$ where $[\mathcal{T}_h]$ is the prior for the hth tree and $[\mathcal{M}_h|\mathcal{T}_h]$ is the collection of leaves for the hth tree
- the collection of leaves for the hth tree are independent: $[\mathcal{M}_h|\mathcal{T}_h] = \prod_n [\mu_{hn}|\mathcal{T}_h]$ where n indexes the leaf nodes

The BART prior

- The tree prior: $[\mathcal{T}_h]$. There are three prior components of $\mathcal{T}_h$ which govern whether the tree branches grow or are pruned
- The first tree prior regularizes the probability of a branch at leaf node n in tree depth $d(n) = \lfloor \log_2 n \rfloor$ as follows.

$$\psi_n \overset{\text{prior}}{\sim} \mathrm{B}(p(d(n))) \text{ where } p(d(n)) = \alpha(d(n)+1)^{-\beta} \quad (1)$$

  $\psi_n = 1$ represents a branch while $\psi_n = 0$ is a leaf
  $0 < \alpha < 1$ and $\beta \geq 0$
- You can specify these prior parameters with arguments, but the following defaults are recommended: α is set by the parameter `base=0.95` and β by `power=2`
- The expected number of branches (leaves) is 1 (2) with probability $\mathbf{P}[\psi_1 = 1, \psi_2 = \psi_3 = 0] = p(0)q(1)^2 \approx 0.55$
- Or 2 (3) with $2\mathbf{P}[\psi_1 = \psi_2 = 1, \psi_3 = \psi_4 = \psi_5 = 0] = 2p(0)p(1)q(1)q(2)^2 \approx 0.27$ (doubled due to symmetry)
- Trees with only 1 or 2 branches (2 or 3 leaves) would dominate with a probability of about **0.82**

BART and Bayesian nonparametric theory

- frequentist theoretical justification for BART's performance: asymptotically consistent with a near optimal learning rate
- the BART posterior distribution concentrates around the truth at a near optimal minimax rate
- the default BART Branching penalty is near optimal:
  $\psi_n \overset{\text{prior}}{\sim} \mathbf{B}\left(\alpha(1+d(n))^{-\beta}\right)$ **where** $d(n) = 0, \ldots$
- the optimal BART Branching penalty is now shown to be:
  $\psi_n \overset{\text{prior}}{\sim} \mathbf{B}\left(\gamma^{d(n)}\right)$ **where** $0 < \gamma < 0.5$

Branches (Leaves)	0 (1)	1 (2)	2 (3)	3+ (4+)
Prior probability	0.00	$(1-\gamma)^2$	$2\gamma(1-\gamma)(1-\gamma^2)^2$	$\ldots$
$\gamma = 0.25$	0.00	0.56	0.33	0.11
$\alpha = 0.95, \beta = 2$	0.05	0.55	0.27	0.13

Chipman, George & McCulloch 1998 *JASA*
Rockova & Saha 2018 *PMLR*
Rockova & van der Pas 2020 *Annals of Statistics*

The BART prior

- The leaf prior: $[\mu_{hn}|\mathcal{T}_h]$
- Given a tree, $\mathcal{T}_h$, there is a prior on its leaf values, $\mu_{hn}|\mathcal{T}_h$ and we denote the collection of all leaves in $\mathcal{T}_h$ by $\mathcal{M}_h = \{(n, \mu_{hn}) : n \in \mathcal{L}_h\}$
- Suppose that $y \in [y_{\min}, y_{\max}]$ where $y_{\min}$ and $y_{\max}$ might be elicited as $\mathbf{0.025}$ and $\mathbf{0.975}$ quantiles otherwise, the observed min and max are used (the default)
- Denote $[\tilde{\mu}_1, \ldots, \tilde{\mu}_H]$ as the leaf output values from each tree corresponding to the vector of covariates, x
- If $\tilde{\mu}_h|\mathcal{T}_h \overset{\text{iid}}{\sim} \mathrm{N}\left(0, \sigma_\mu^2\right)$, then the model estimate is $\hat{y} = \mathrm{E}[y|x] = \mu + \sum_h \tilde{\mu}_h$ where $\hat{y} \sim \mathrm{N}\left(\mu, H\sigma_\mu^2\right)$
- Solve for σ_μ: $y_{\min} = \mu - \kappa\sqrt{H}\sigma_\mu$ and $y_{\max} = \mu + \kappa\sqrt{H}\sigma_\mu$ $\sigma_\mu = \frac{y_{\max} - y_{\min}}{2\kappa\sqrt{H}}$
- Therefore, we arrive at $\mu_{hn} \overset{\text{prior}}{\sim} \mathrm{N}\left(0, \frac{\tau^2}{4H\kappa^2}\right)$ **where** $\tau = y_{\max} - y_{\min}$

The BART prior

- The parameter κ calibrates this prior as follows
- The default value, $\kappa = 2$, corresponds to $\hat{y}$ falling within the extrema with approximately 0.95 probability
- Alternative choices of κ can be supplied via the `k` argument
- We have found that values of $\kappa \in [1, 3]$ generally yield good results
- Note that κ is a potential candidate parameter for choice via cross-validation

The BART prior

- We fix the number of trees at $\boldsymbol{H}$ which corresponds to the argument `ntree`
- The default number of trees is 200 for continuous outcomes; but for computational convenience, 50 is also a reasonable choice which is the default for all other outcomes
- cross-validation could be considered

The BART prior

- The number of cut-points is provided by the argument `numcut` and the default is 100
- The default number of cut-points is achieved for continuous covariates
- For continuous covariates, the cut-points are uniformly distributed by default, or generated via uniform quantiles if the argument `usequants=TRUE` is provided
- By default, discrete covariates which have fewer than 100 values will necessarily have fewer cut-points
- However, if you want a single discrete covariate to be represented by a group of binary dummy variables, one for each category, then pass the variable as a factor within a data frame

The BART prior

- Next, there is a prior dictating the choice of a splitting variable j conditional on a branch event ψ_n which defaults to uniform probability $s_j = P^{-1}$ where P is the number of covariates
- Given a branch event, ψ_n, and a variable chosen, x_j, the last tree prior selects a cut point, c_{jk}, within the range of observed values for x_j; this prior is uniform

The BART error variance prior: $[\sigma^2]$

- The prior for σ^2 is the conjugate scaled inverse Chi-square distribution, i.e., $\lambda\nu\chi^{-2}(\nu)$
- Equivalent to the inverse Gamma, i.e., $\textbf{Gamma}^{-1}(\nu/2,\ \lambda\nu/2)$ **where** $\mathrm{E}[\sigma^2] = \lambda\nu/(\nu-2)$
- We recommend that the degrees of freedom, ν, be from 3 to 10 and the default is 3 (can be over-ridden by the argument `sigdf`)
- The λ parameter can be specified by the `lambda` argument (defaults to `NA`)
- If `lambda` is unspecified, then we determine a reasonable value for λ based on an estimate, $\widehat{\sigma}$, (which can be specified by the argument `sigest` and defaults to `NA`)

The BART error variance prior

- If `sigest` is unspecified, the default value of `sigest` is determined via linear regression or the sample standard deviation: if $P < N$, then $y_i \sim \mathrm{N}\left(x_i'\widehat{\beta}, \widehat{\sigma}^2\right)$; otherwise, $\widehat{\sigma} = s_y$
- Now we solve for λ such that $\mathrm{P}\left[\sigma^2 \leq \widehat{\sigma}^2\right] = q$
- This quantity, q, can be specified by the argument `sigquant` and the default is 0.9 whereas we also recommend considering 0.75 and 0.99
- Note that the pair (ν, q) are potential candidate parameters for choice via cross-validation.

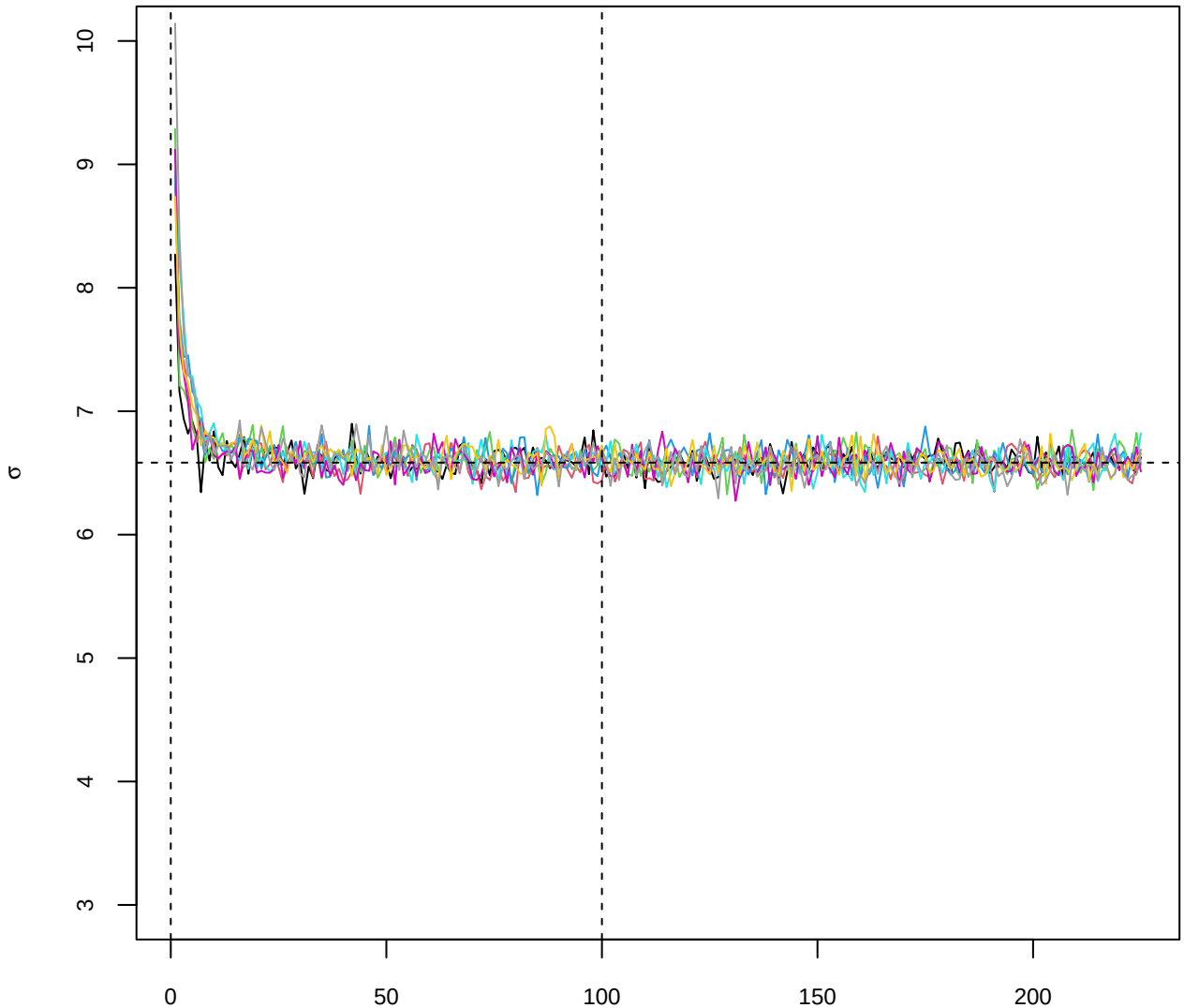
Returning to the real data example

- The CDC mainly used the US National Health and Nutrition Examination Survey (NHANES): waves I-III $\boldsymbol{n = 12677}$
- For simplicity, I used NHANES 1999-2000 annual/continuous
- The data set is in the BART3 package: `bmx`
  and see the `height3.R` example in `demo`
- 2-17 years (fractional age for months)
- each child only measured once
- height (cm) and weight (kg) collected
- Check MCMC convergence with $\boldsymbol{\max \widehat{R} < 1.1}$ for σ:
  Vehtari, Gelman et al. 2021 *Bayesian Analysis*

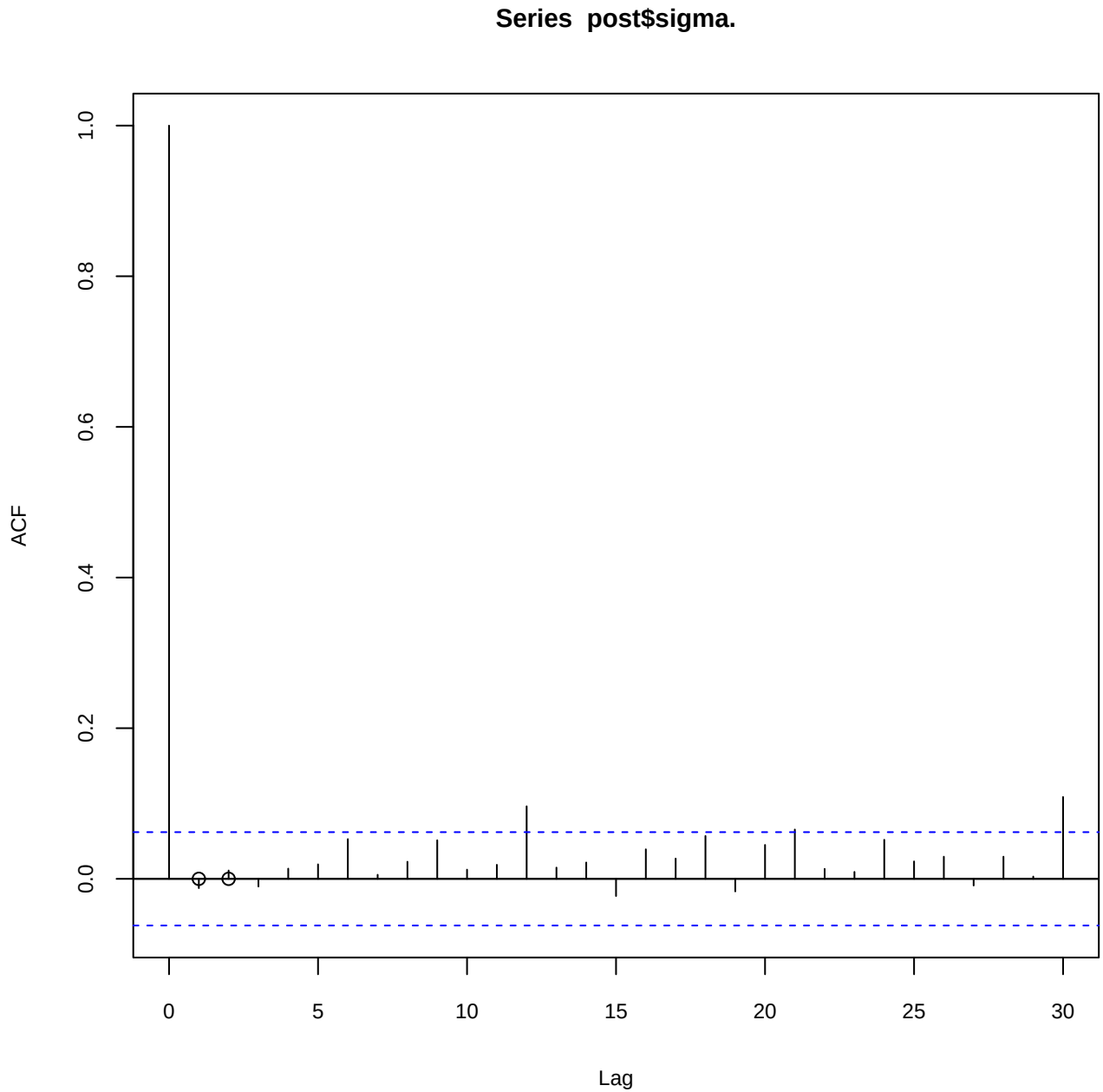
	n	%
Total	3435	
Males	1768	51.5
Females	1667	48.5
White	800	23.3
Black	1035	30.1
Hispanic	1600	46.6

MCMC Convergence `post$sigma`: $\max \widehat{R} = 1.01$

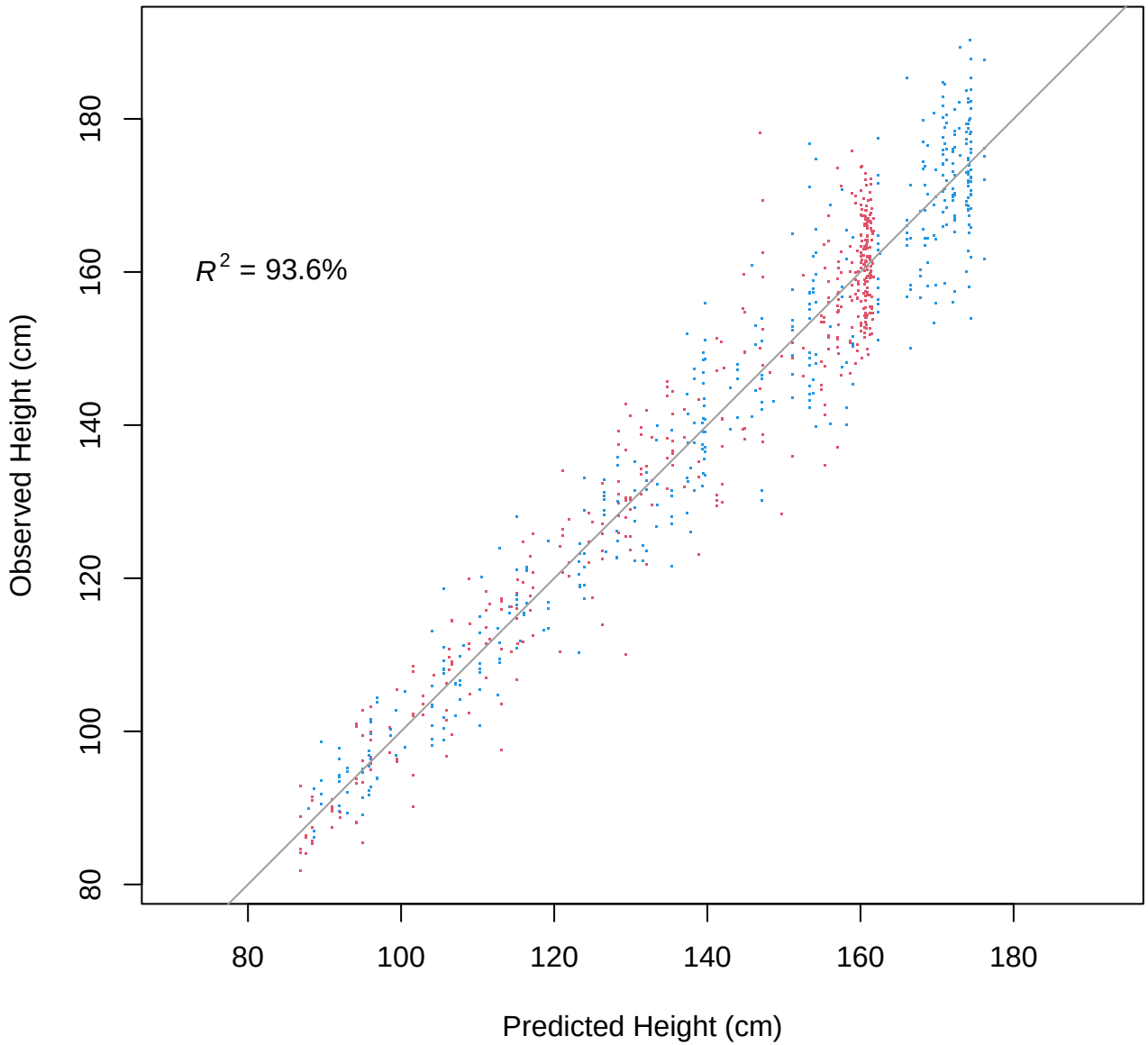
Burn-in 100, Thinning 1, Chains 8, Posterior 1000



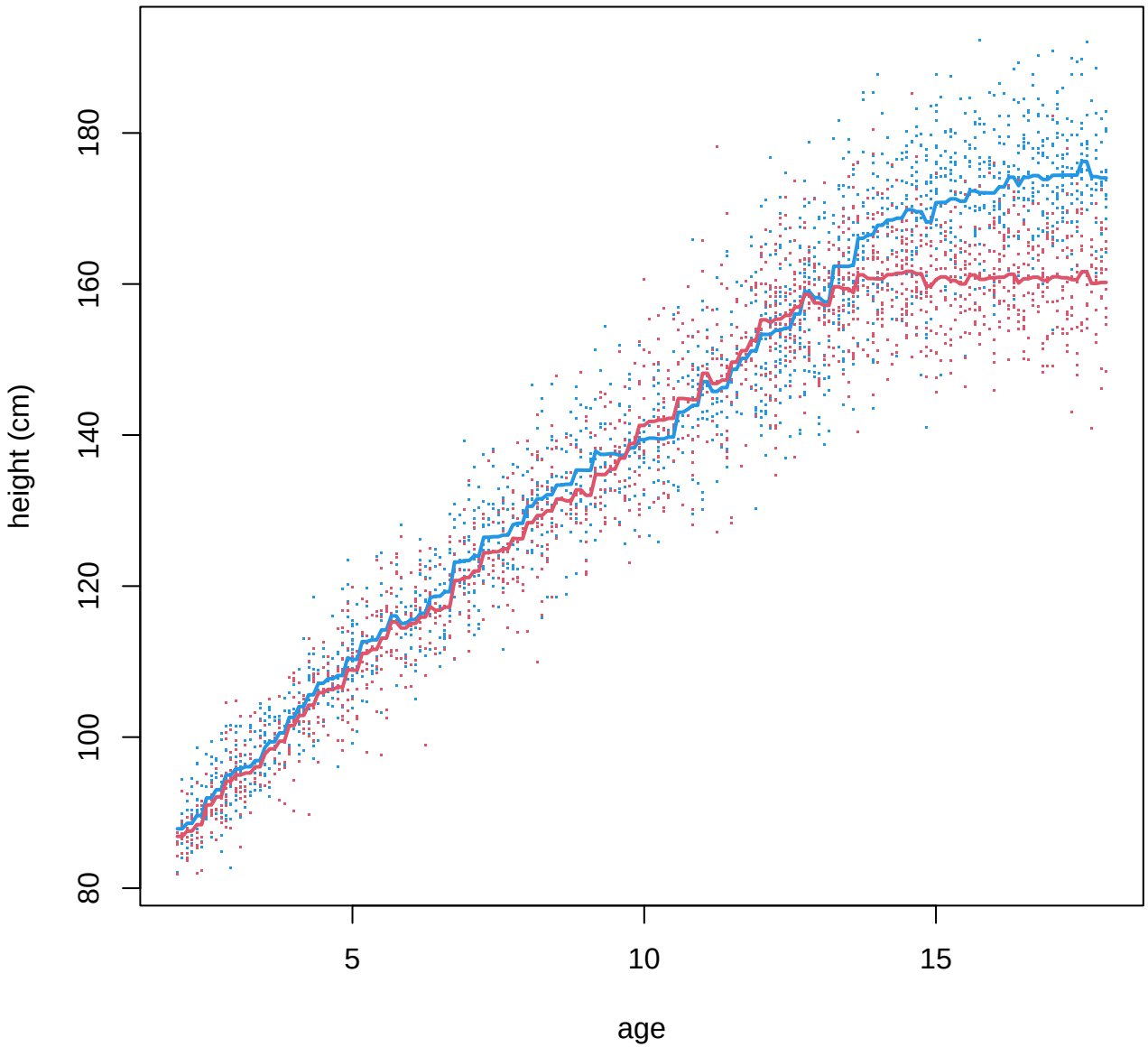
MCMC Convergence `post$sigma`: Auto-correlation



Test data: M vs. F



Data fit: M vs. F



95% credible intervals: M vs. F