STATISTICAL AND MACHINE LEARNING FOR BIG GEOSPATIAL DATA: Part II

Abhi Datta
Johns Hopkins University
Department of Biostatistics

Overview of Part II

Spatial Linear Models
- Limitations of linearity

Non-linear regression methods for spatial data
- Basis functions and GAMs
- Machine learning methods like Random forests (RF) and Neural Networks (NN)

Issues of standard random forests for spatial or time-series data

RF-GLS: Random forests for spatial data with explicit modeling of spatial correlation
- Spatial and time-series examples

Demonstration of RandomForestsGLS R-package implementing RFGLS

Spatial linear mixed effects model

SLMM: $Y_i = X_i'\beta + w_i + \epsilon_i^*, w \sim GP(0,C), \epsilon^* \sim_{iid} N(0,\tau^2)$

Dependent errors: $Y_i = X_i'\beta + \epsilon_i, \epsilon_i = w_i + \epsilon_i^*$

The errors ϵ_i are now a dependent process, $Cov(\epsilon_i, \epsilon_j) = C_{ij} + \tau^2 I(i = j)$

Marginal model: $Y \sim N(X\beta, \Sigma)$ where $\Sigma = Cov(\epsilon) = C(\theta) + \tau^2 I$

Spatial linear mixed effects model

SLMM: $Y_i = X_i'\beta + w_i + \epsilon_i^*, w \sim GP(0,C), \epsilon^* \sim_{iid} N(0,\tau^2)$

Dependent errors: $Y_i = X_i'\beta + \epsilon_i, \epsilon_i = w_i + \epsilon_i^*$

The errors ϵ_i are now a dependent process, $Cov(\epsilon_i, \epsilon_j) = C_{ij} + \tau^2 I(i = j)$

Marginal model: $Y \sim N(X\beta, \Sigma)$ where $\Sigma = Cov(\epsilon) = C(\theta) + \tau^2 I$

Linearity is a strong assumption on the relationship between the response and covariates

Non-linear models for dependent data

Dependent errors: $Y_i = X_i'\beta + \epsilon_i$

The errors ϵ_i are a dependent process, $Cov(\epsilon_i, \epsilon_j) = C_{ij} + \tau^2 I(i = j)$

Non-linear models for dependent data

Dependent errors: $Y_i = \cancel{X_i'\beta}\, m(X_i) + \epsilon_i$

The errors ϵ_i are a dependent process, $Cov(\epsilon_i, \epsilon_j) = C_{ij} + \tau^2 I(i = j)$

Non-linear mean function $E(Y_i) = m(X_i)$

Reduces to the linear model when $m(X_i) = X_i'\beta$

Classic non-linear models for dependent data

Basis functions (Diggle and Hutchinson, 1989)

$E(Y_i) = m(X_i) = B(X_i)\gamma$, $B(X_i)$ are basis functions in X_i

Marginal model: $Y \sim N(B(X)\gamma, \Sigma)$

Still a linear model in the regression coefficients γ

Can be implemented in the same way as the spatial linear model

Classic non-linear models for dependent data

Basis functions (Diggle and Hutchinson, 1989)

$E(Y_i) = m(X_i) = B(X_i)\gamma$, $B(X_i)$ are basis functions in X_i

Marginal model: $Y \sim N(B(X)\gamma, \Sigma)$

Still a linear model in the regression coefficients γ

Can be implemented in the same way as the spatial linear model

Basis functions directly on the multivariate X_i

Suffers from curse of dimensionality when X_i is more than 2- or 3-dimensional (Taylor and Einbeck, 2013)

Classic non-linear models for dependent data

GAMs (generalized additive models) for spatial data (Nandy et al., 2017, JRSSB)

$$E(Y_i) = m(X_i) = \sum_{j=1}^{d} m_j(X_{ij})$$

Each m_j represented as basis functions

Reduces to special case of basis function models

Classic non-linear models for dependent data

GAMs (generalized additive models) for spatial data (Nandy et al., 2017, JRSSB)

$$E(Y_i) = m(X_i) = \sum_{j=1}^{d} m_j(X_{ij})$$

Each m_j represented as basis functions

Reduces to special case of basis function models

GAMs do not model interactions

Machine learning for dependent data

ML algorithms like random forests (RF, Breiman) and neural nets (NN) can model higher order interactions

RF and NN can approximate any smooth function (Universal approximation result for NN, Hornik, Stinchcombe, White, 1989)

Asymptotic theory supporting Breiman's random forests (Scornet et al. 2015)

Asymptotic theory on neural nets working better than basis functions (Schmidt-Hieber, 2020)

Machine learning for dependent data

Environmental Pollution
Volume 277, 15 May 2021, 116846

Using a land use regression model with machine learning to estimate ground level $PM_{2.5}$ ☆

Atmospheric Environment
Volume 191, October 2018, Pages 205-213

Spatial estimation of urban air pollution with the use of artificial neural network models

Highlights

- ANN models are superior compared to MLR for air pollution spatial forecasting.

Environmental Science & Technology > Vol 51/Issue 12 > Article

ARTICLE | May 23, 2017

Estimating $PM_{2.5}$ Concentrations in the Conterminous United States Using the Random Forest Approach

Rapid rise in use of machine learning algorithms for geospatial analysis

Machine learning for dependent data

Environmental Pollution
Volume 277, 15 May 2021, 116846

Using a land use regression model with machine learning to estimate ground level $PM_{2.5}$ ☆

Atmospheric Environment
Volume 191, October 2018, Pages 205-213

Spatial estimation of urban air pollution with the use of artificial neural network models

Highlights

- ANN models are superior compared to MLR for air pollution spatial forecasting.

Environmental Science & Technology > Vol 51/Issue 12 > Article

ARTICLE | May 23, 2017

Estimating $PM_{2.5}$ Concentrations in the Conterminous United States Using the Random Forest Approach

Rapid rise in use of machine learning algorithms for geospatial analysis

Impact of ignoring data correlation in random forests

ML function classes are non-linear in the parameters

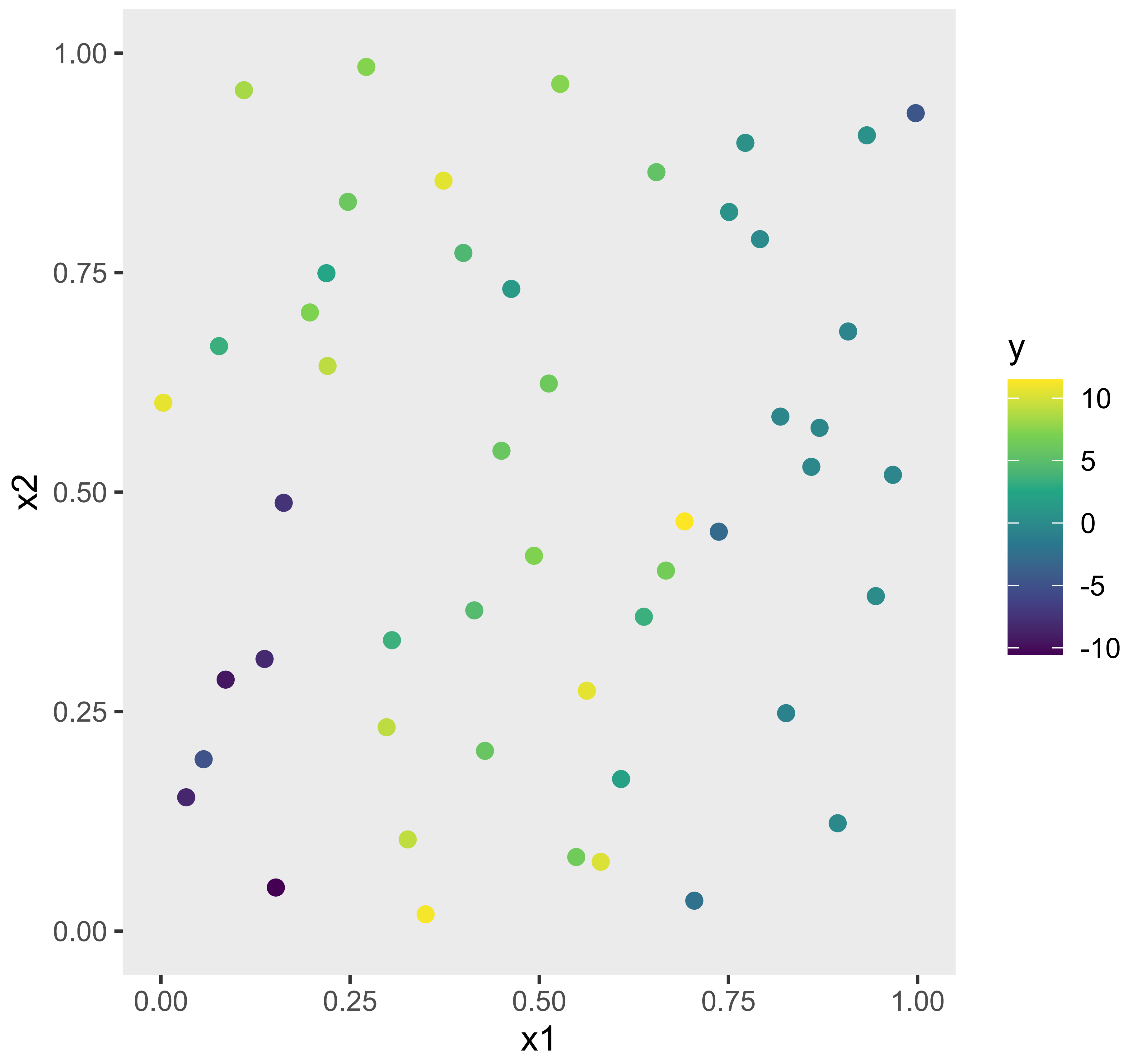
Until recently, most ML algorithms could not directly account for correlation for dependent (spatial/time series) data

What is the impact of ignoring data correlation?

How to use RF or NN while explicitly modeling data correlation?

Review of Regression Tress and Random Forests

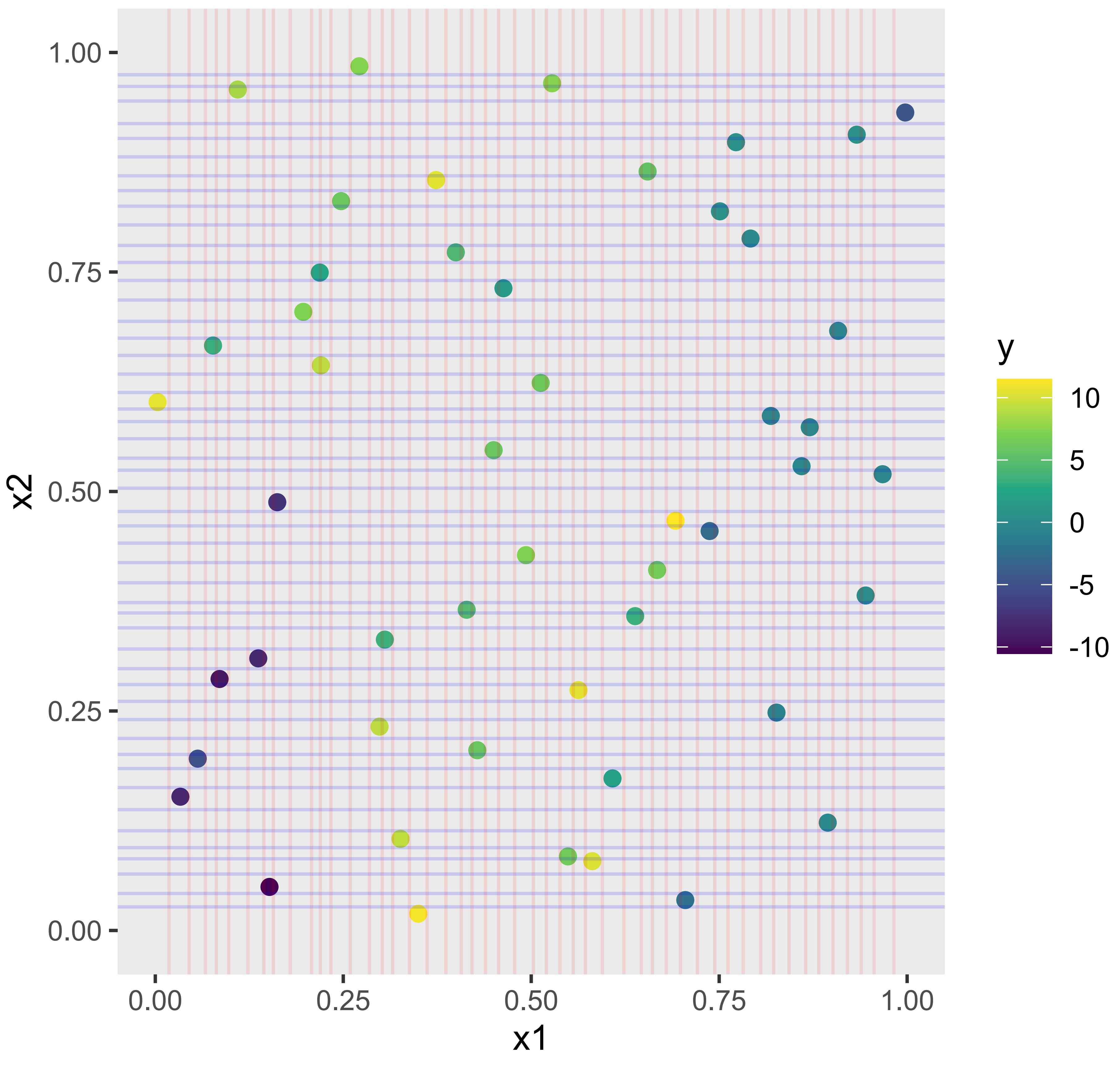
Regression trees



$[0,1]^2$

Review of Regression Tress and Random Forests

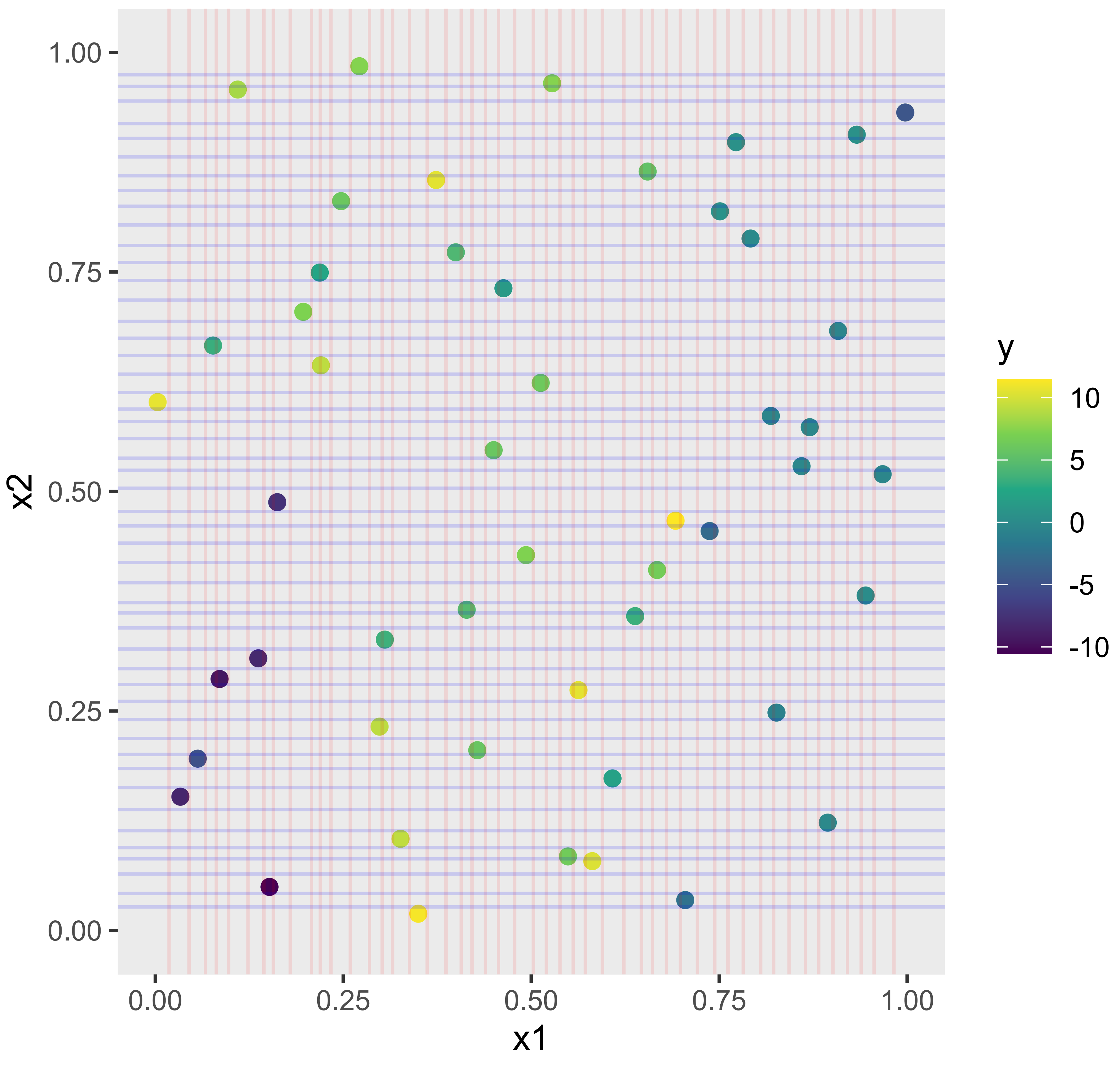
Regression trees



$[0,1]^2$

Review of Regression Tress and Random Forests

Regression trees

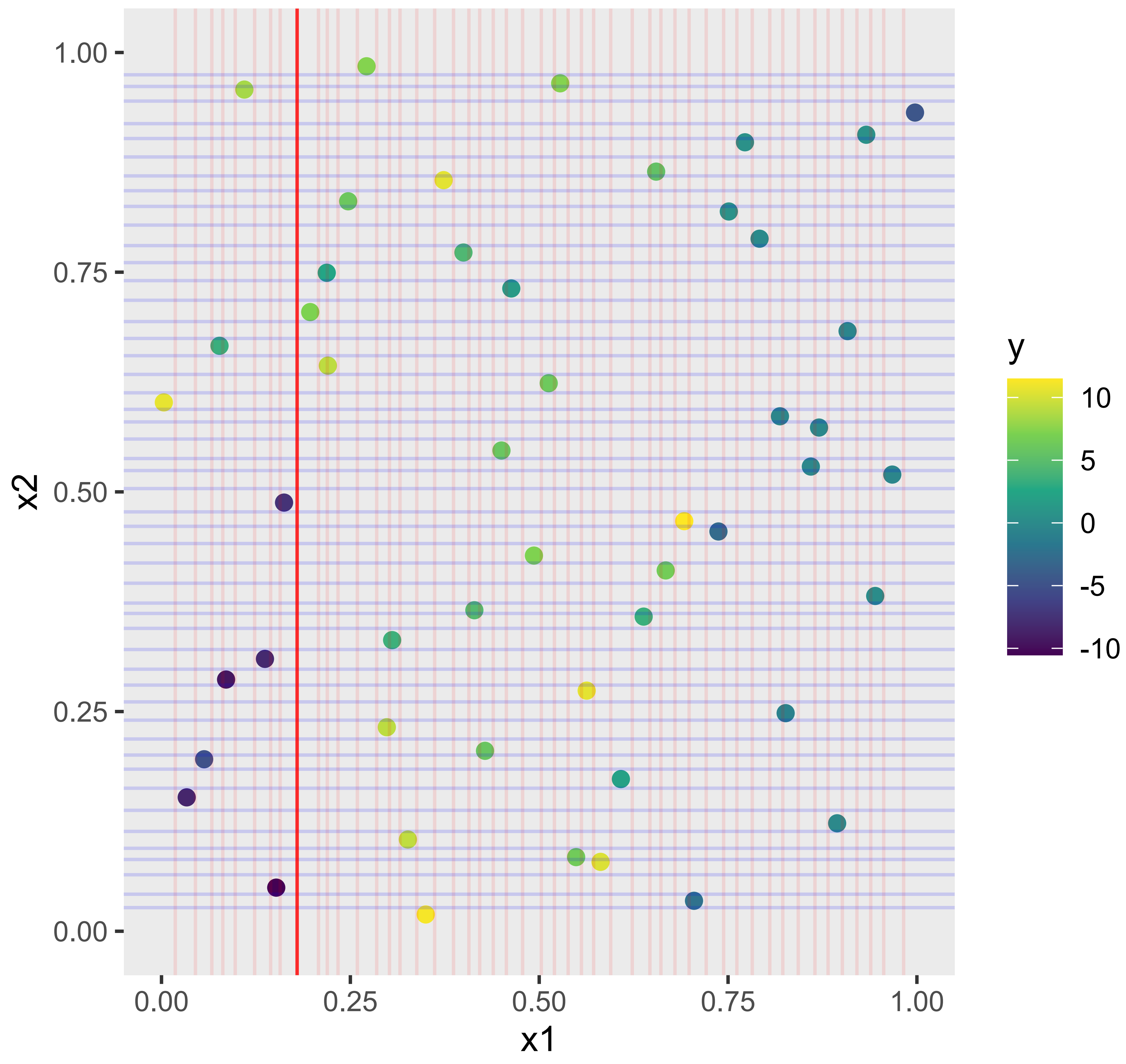


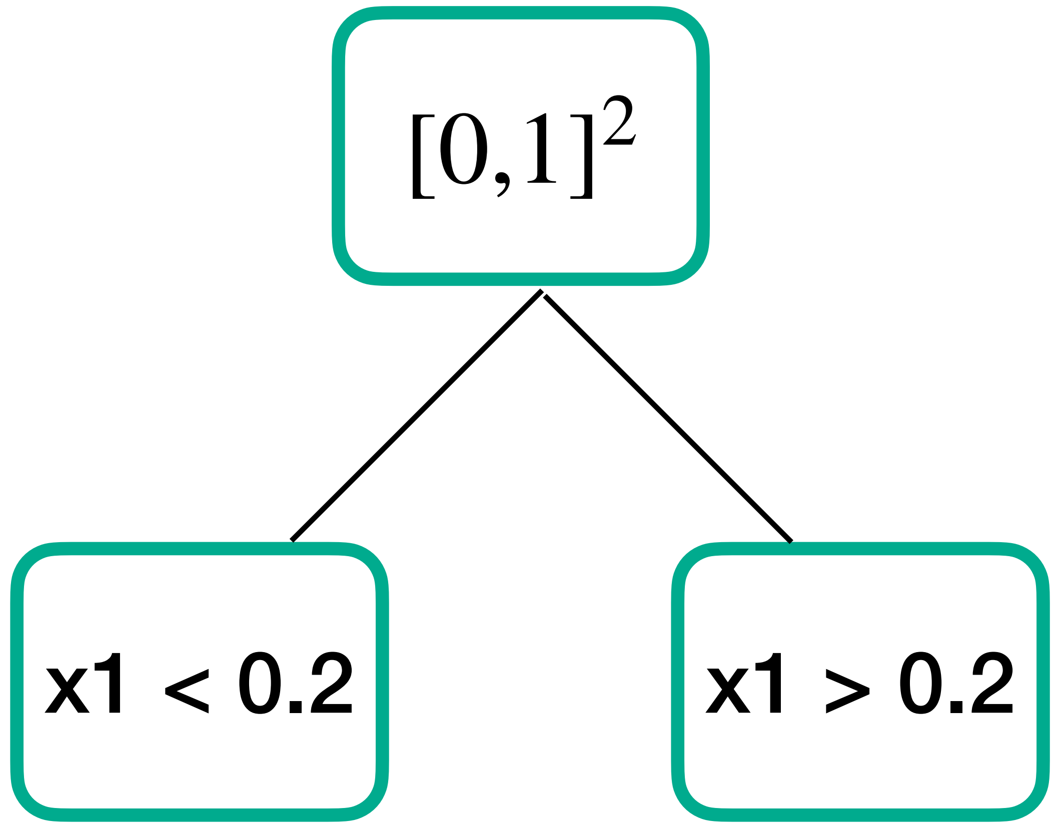
$[0,1]^2$

CART (Classification and regression tree) Split criterion: Maximize $\frac{1}{n_P}\left[\sum_{i=1}^{n_P}(Y_i^P-\bar{Y}^P)^2-\sum_{i_r=1}^{n_R}(Y_i^R-\bar{Y}^R)^2-\sum_{i_l=1}^{n_L}(Y_i^L-\bar{Y}^L)^2\right]$

Review of Regression Tress and Random Forests

Regression trees

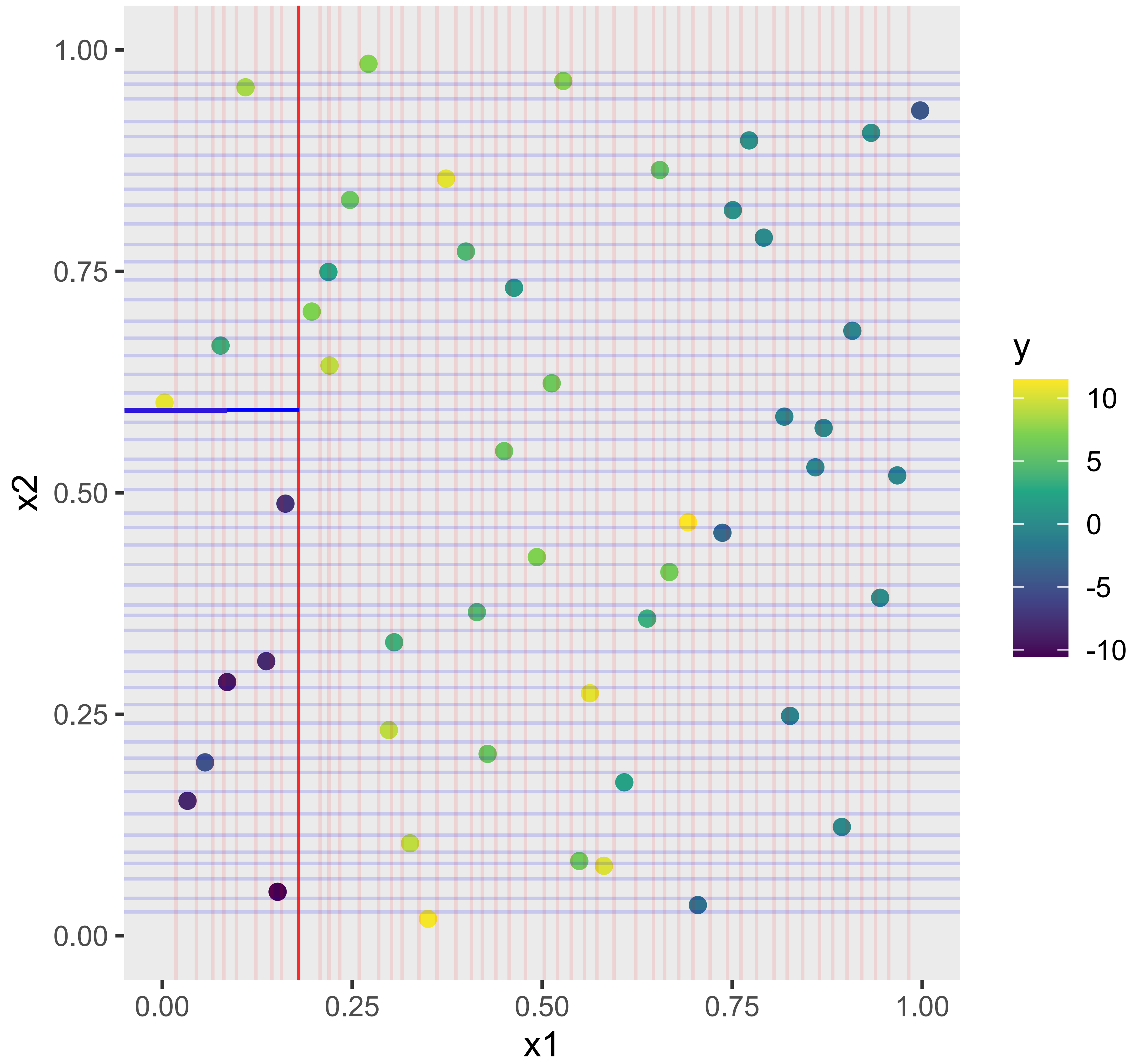


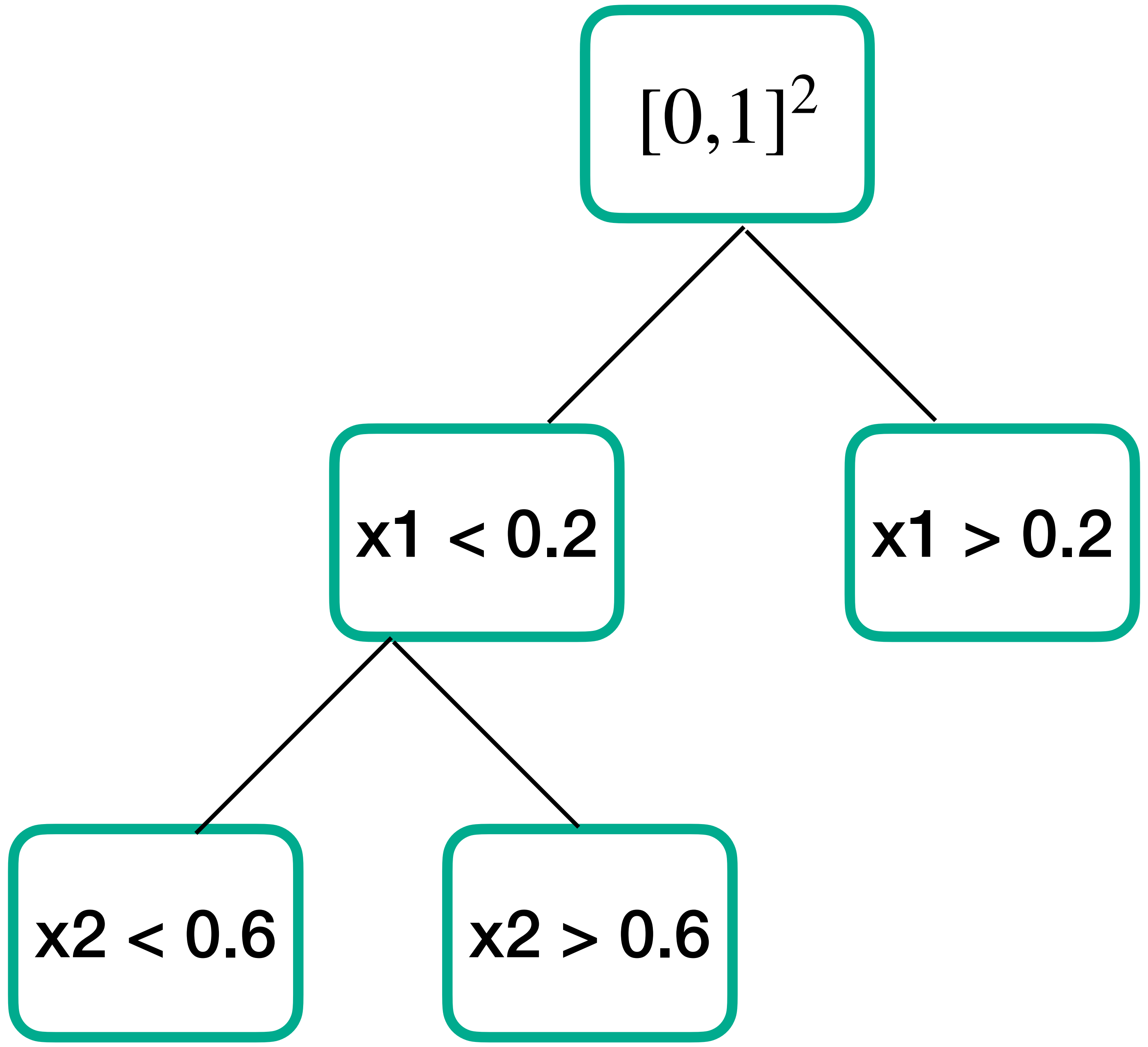


CART (Classification and regression tree) Split criterion: Maximize $\frac{1}{n_P}\left[\sum_{i=1}^{n_P}(Y_i^P-\bar{Y}^P)^2-\sum_{i_r=1}^{n_R}(Y_i^R-\bar{Y}^R)^2-\sum_{i_l=1}^{n_L}(Y_i^L-\bar{Y}^L)^2\right]$

Review of Regression Tress and Random Forests

Regression trees

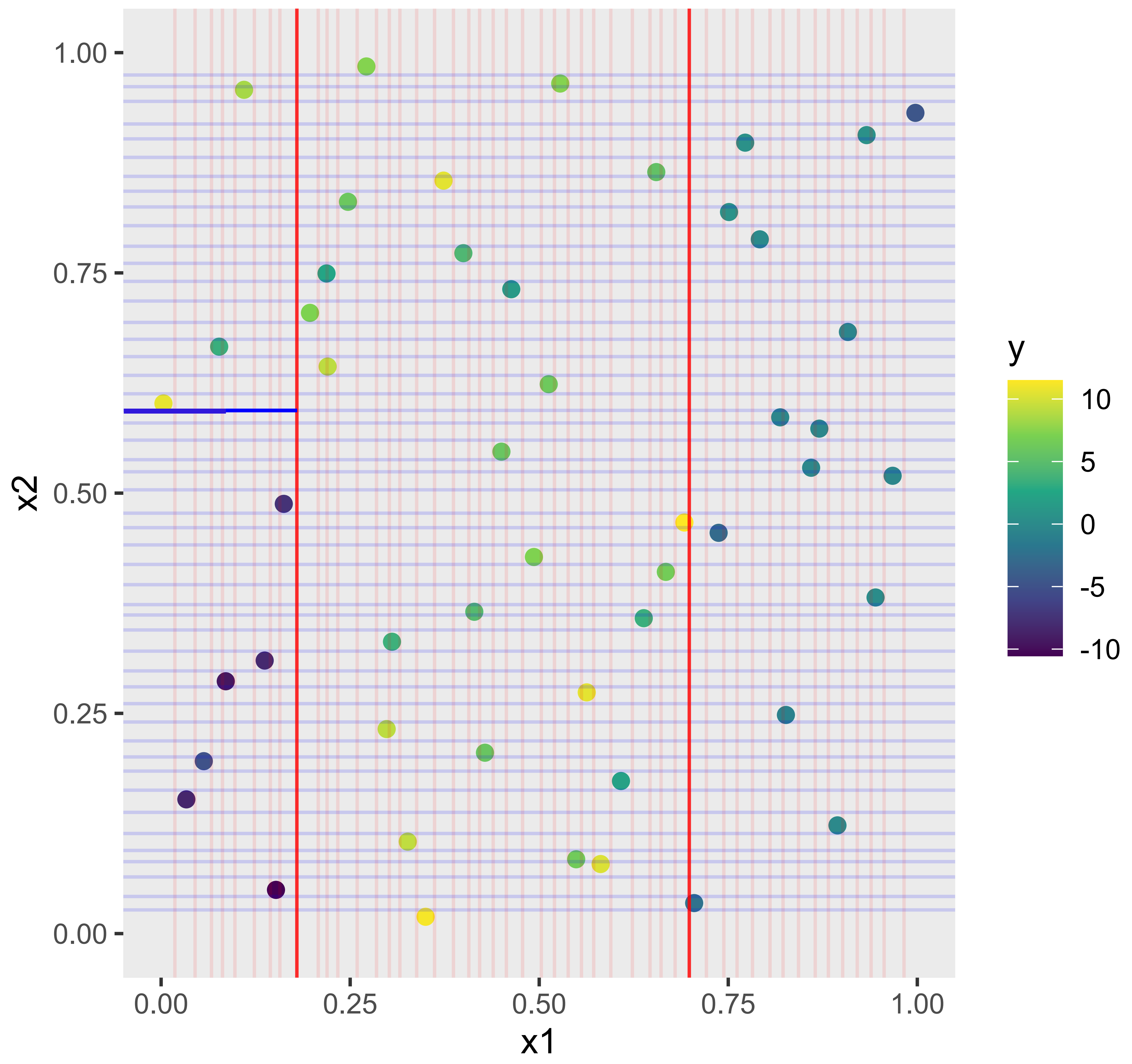


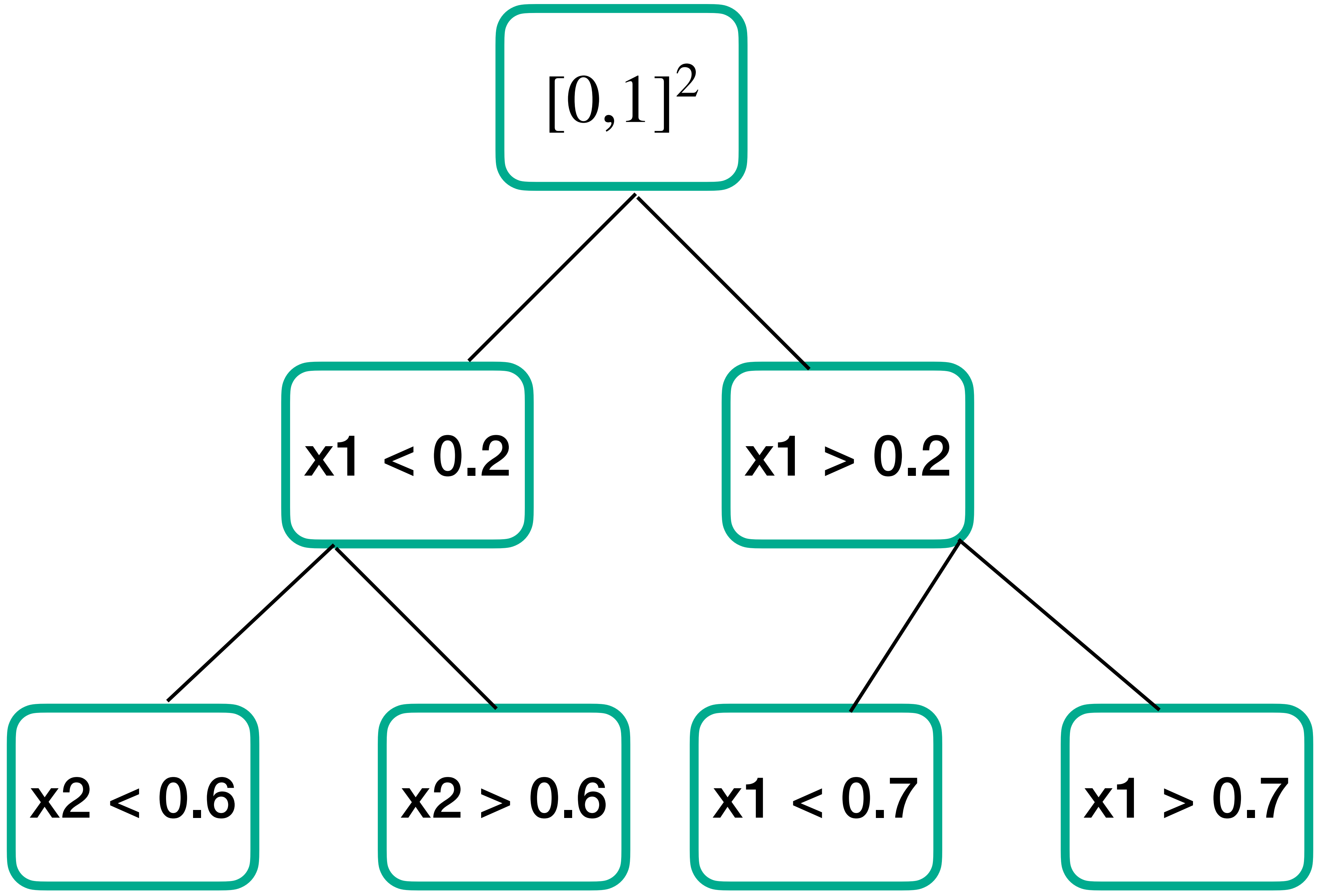


CART (Classification and regression tree) Split criterion: Maximize $\frac{1}{n_P}\left[\sum_{i=1}^{n_P}(Y_i^P - \bar{Y}^P)^2 - \sum_{i_r=1}^{n_R}(Y_i^R - \bar{Y}^R)^2 - \sum_{i_l=1}^{n_L}(Y_i^L - \bar{Y}^L)^2\right]$

Review of Regression Tress and Random Forests

Regression trees





CART (Classification and regression tree) Split criterion: Maximize $\frac{1}{n_P}\left[\sum_{i=1}^{n_P}(Y_i^P-\bar{Y}^P)^2-\sum_{i_r=1}^{n_R}(Y_i^R-\bar{Y}^R)^2-\sum_{i_l=1}^{n_L}(Y_i^L-\bar{Y}^L)^2\right]$

Review of Regression Tress and Random Forests

Data: $(Y_i, X_i) \in \mathbb{R} \times \mathbb{R}^d, i = 1,\ldots,n$

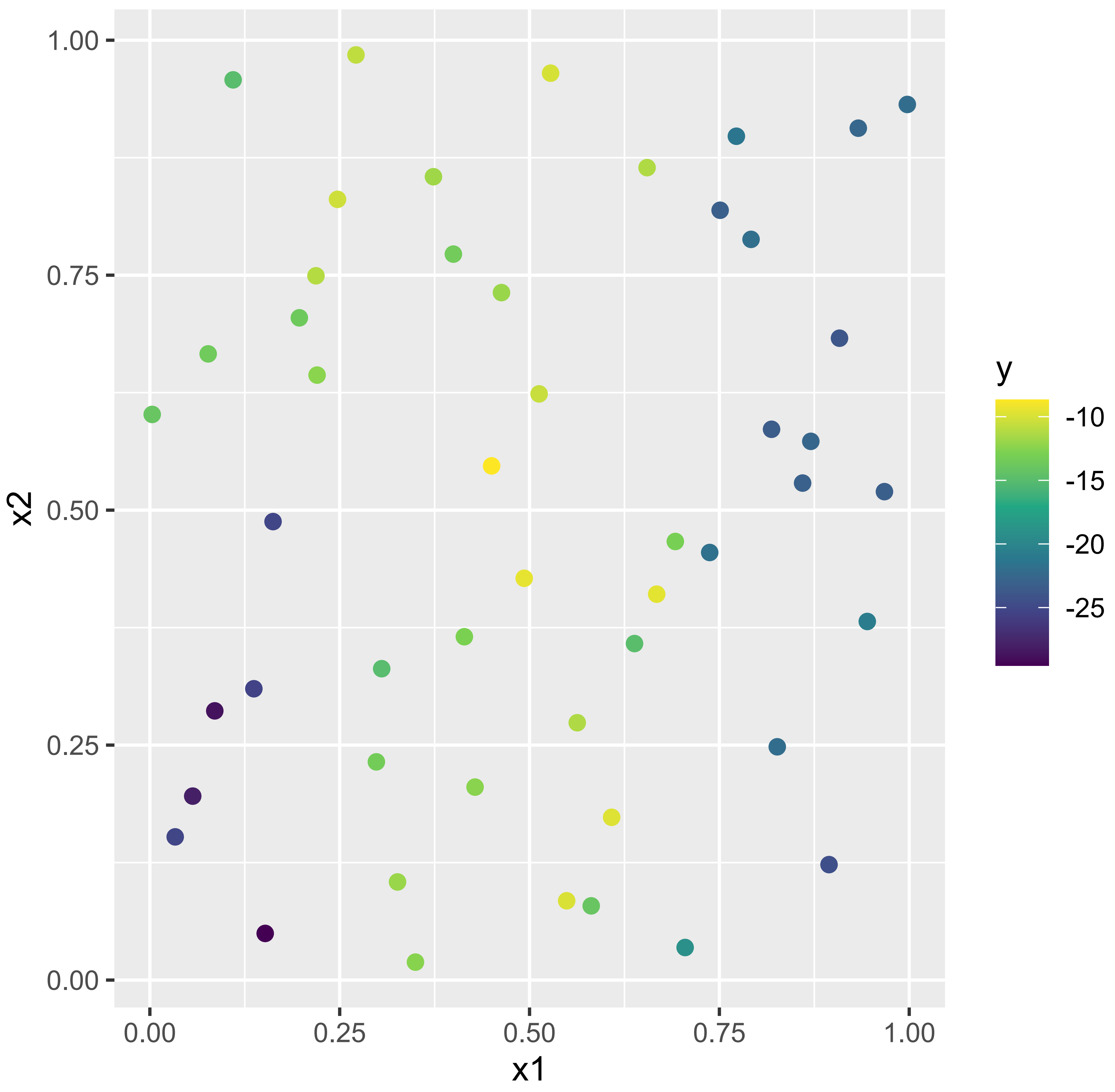
Node creation: Sequentially maximizing the CART-split criterion within each node

Representative assignment: The value of the tree estimator at each leaf node is the mean of the responses of the node members

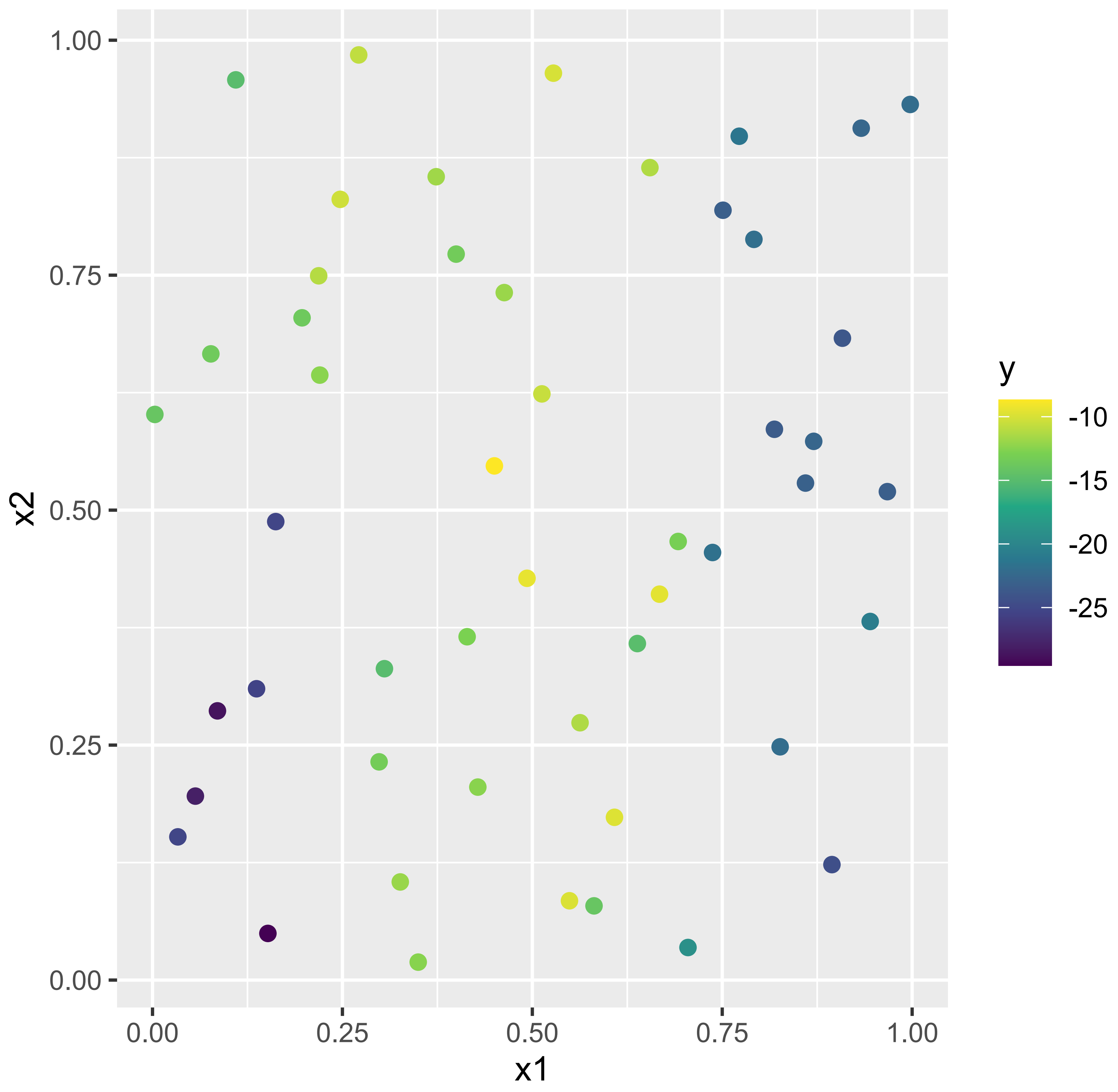
Random feature selection: For each split, only consider a randomly chosen (mtry << D) subset of the features as candidate split direction

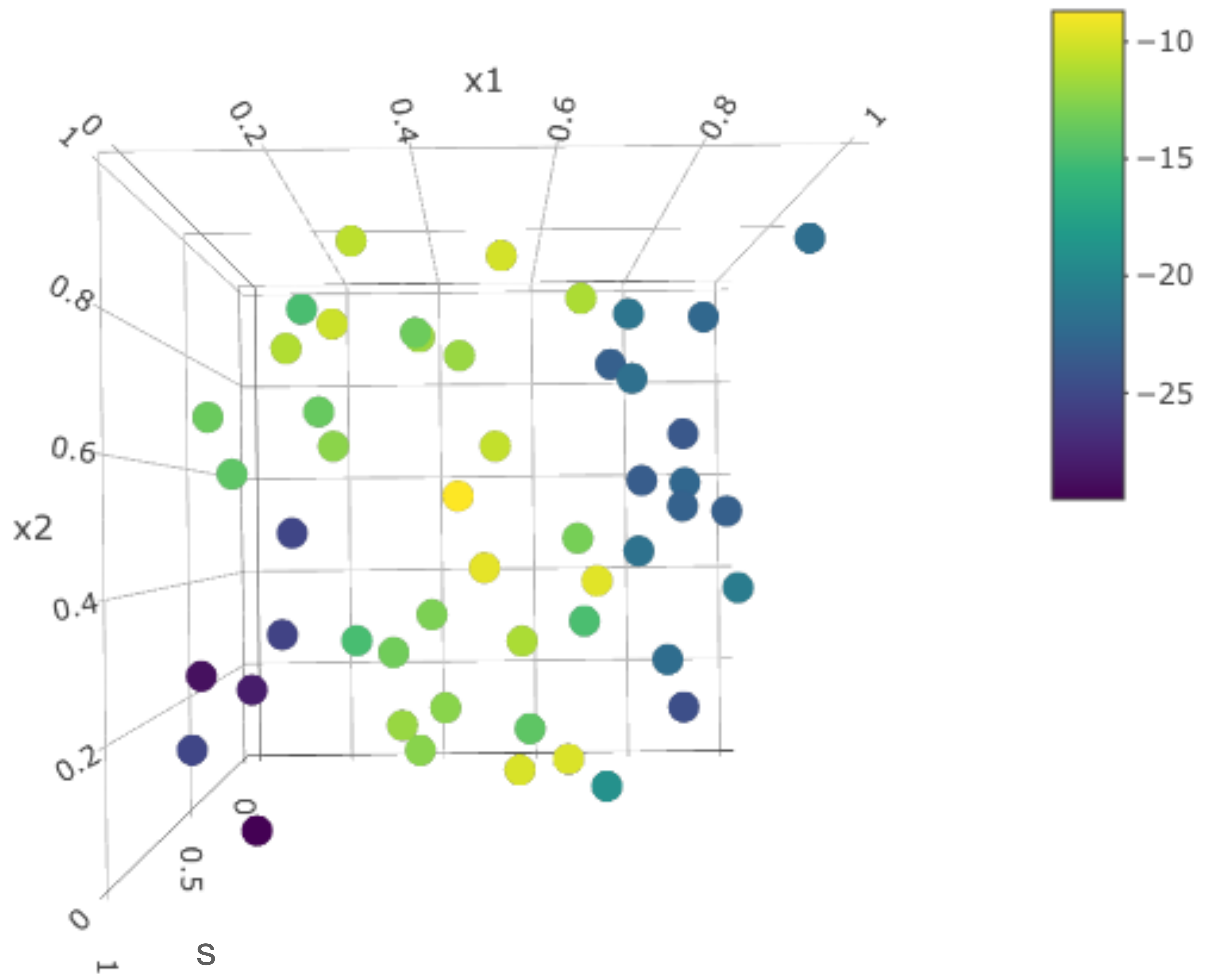
Bagging / subsampling: RF estimate = average of a large number of regression trees, each tree grown with a resample/subsample of the data

Issues of RF for dependent data

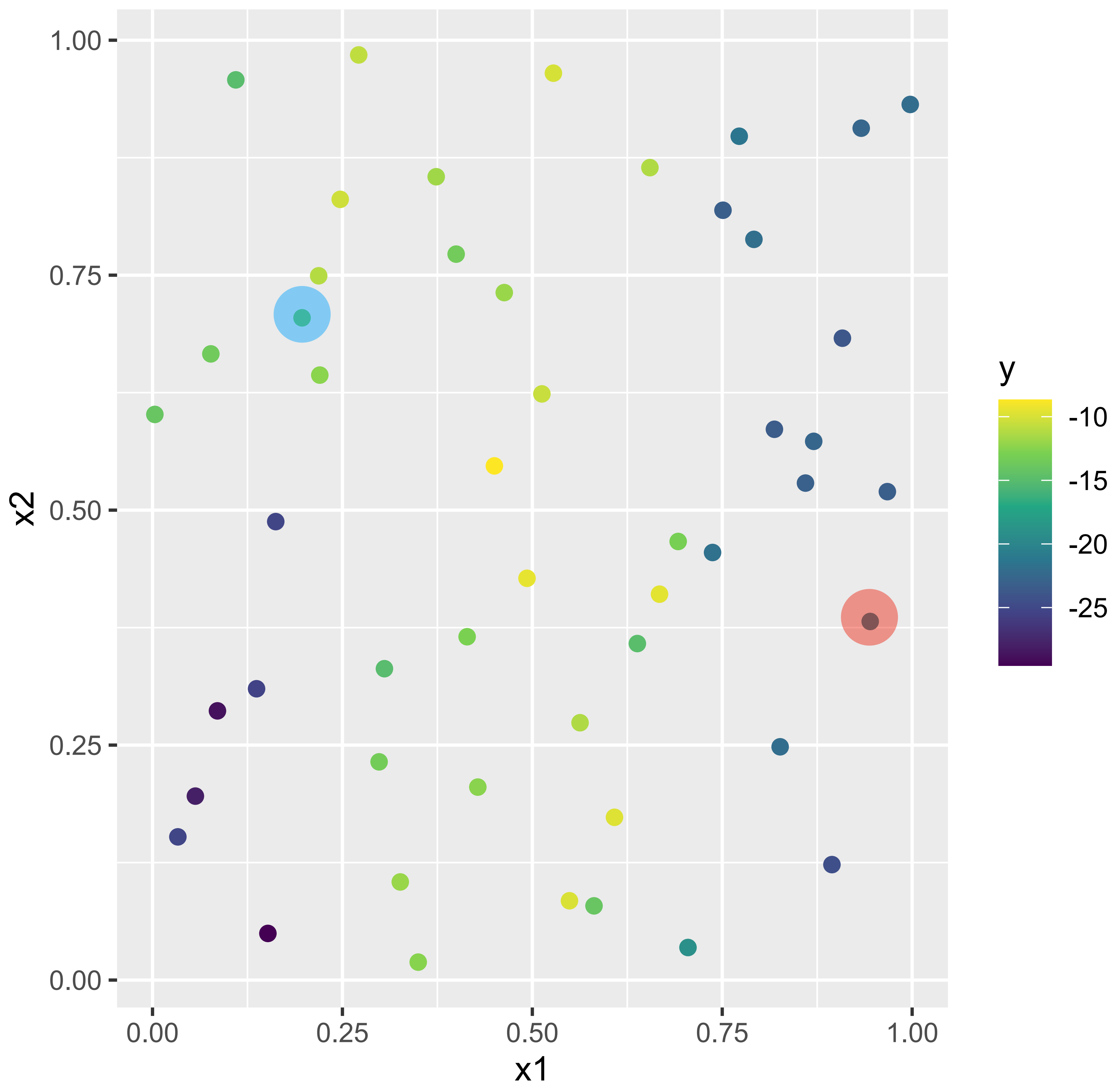


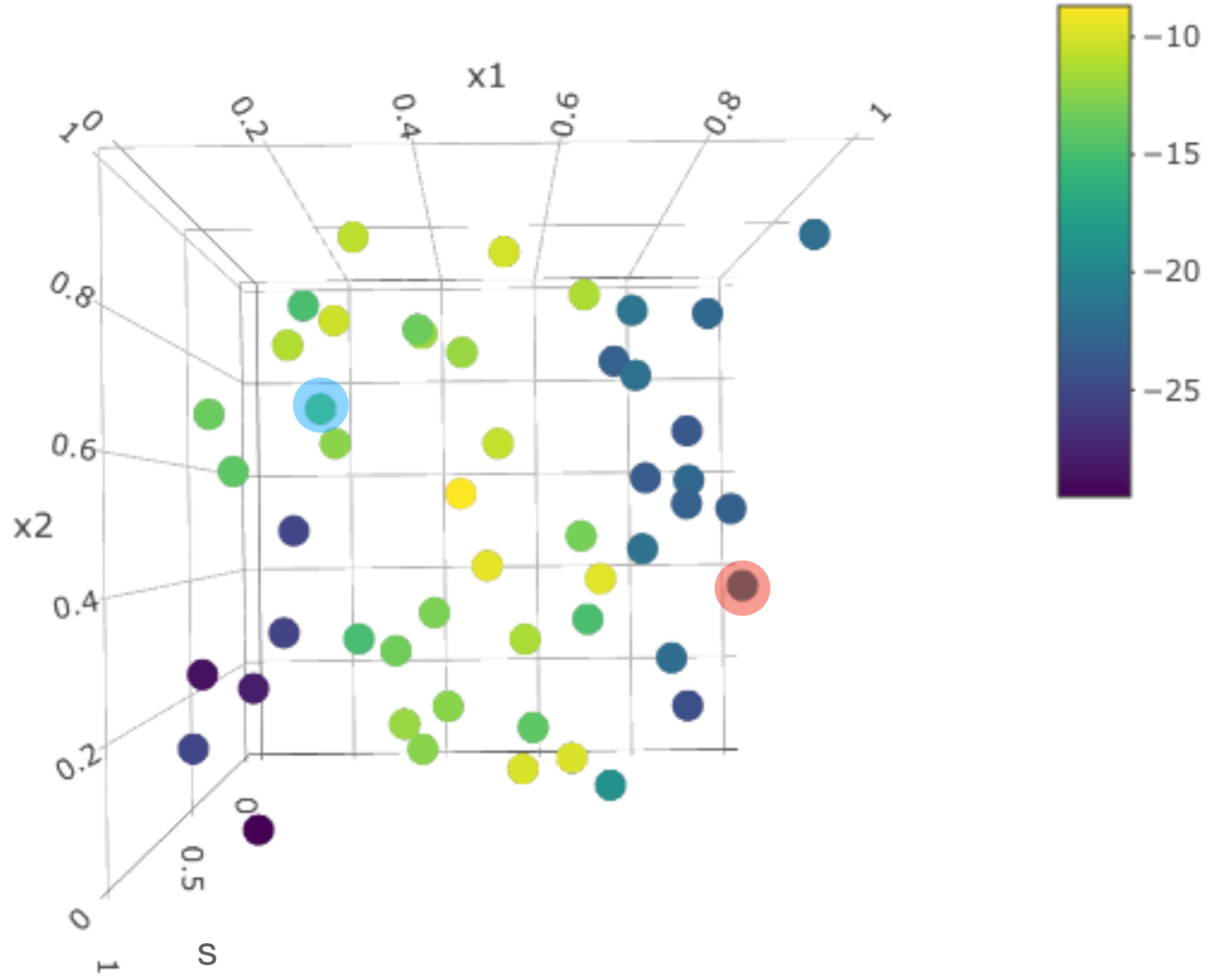
Issues of RF for dependent data





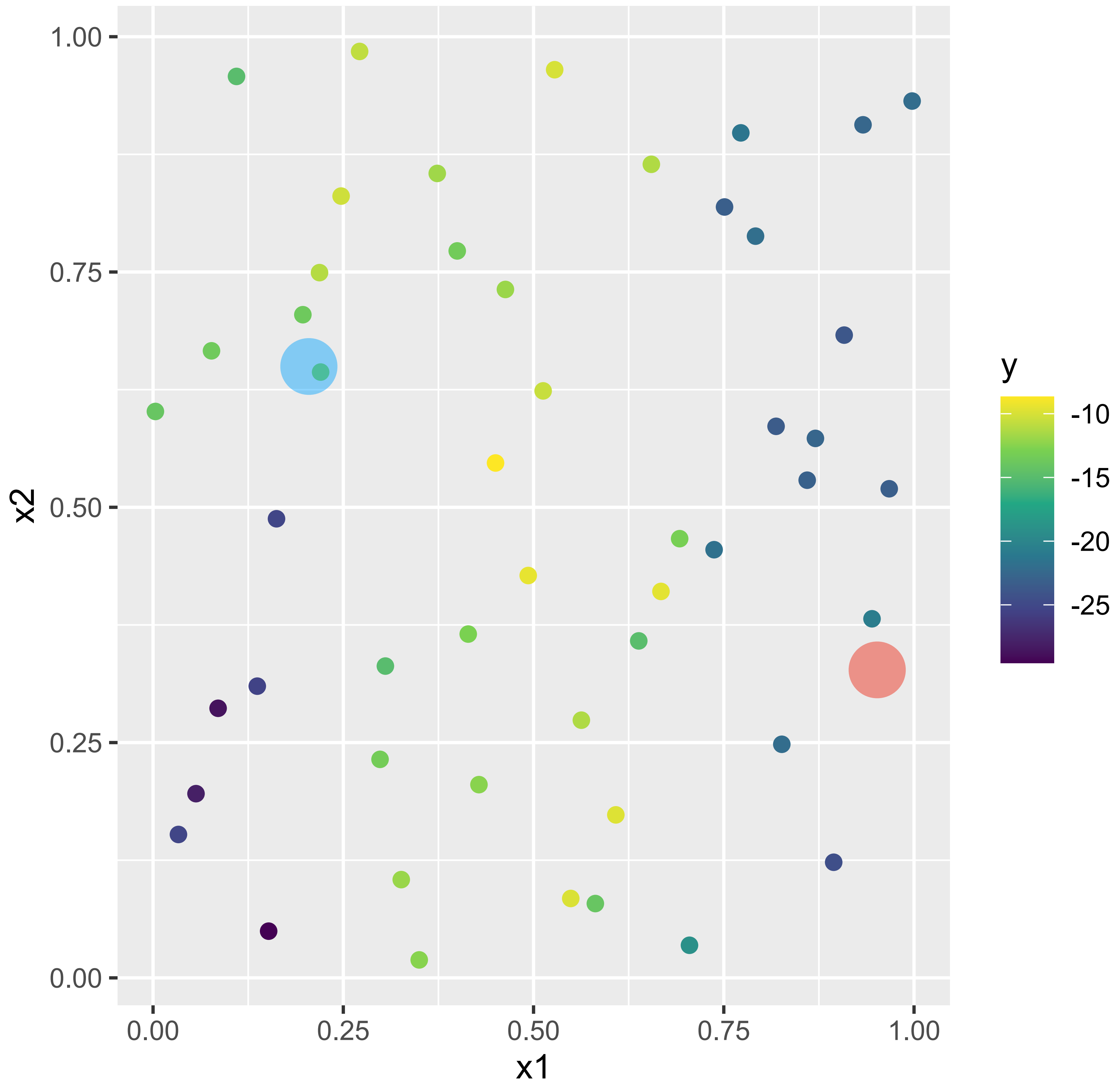
Issues of RF for dependent data

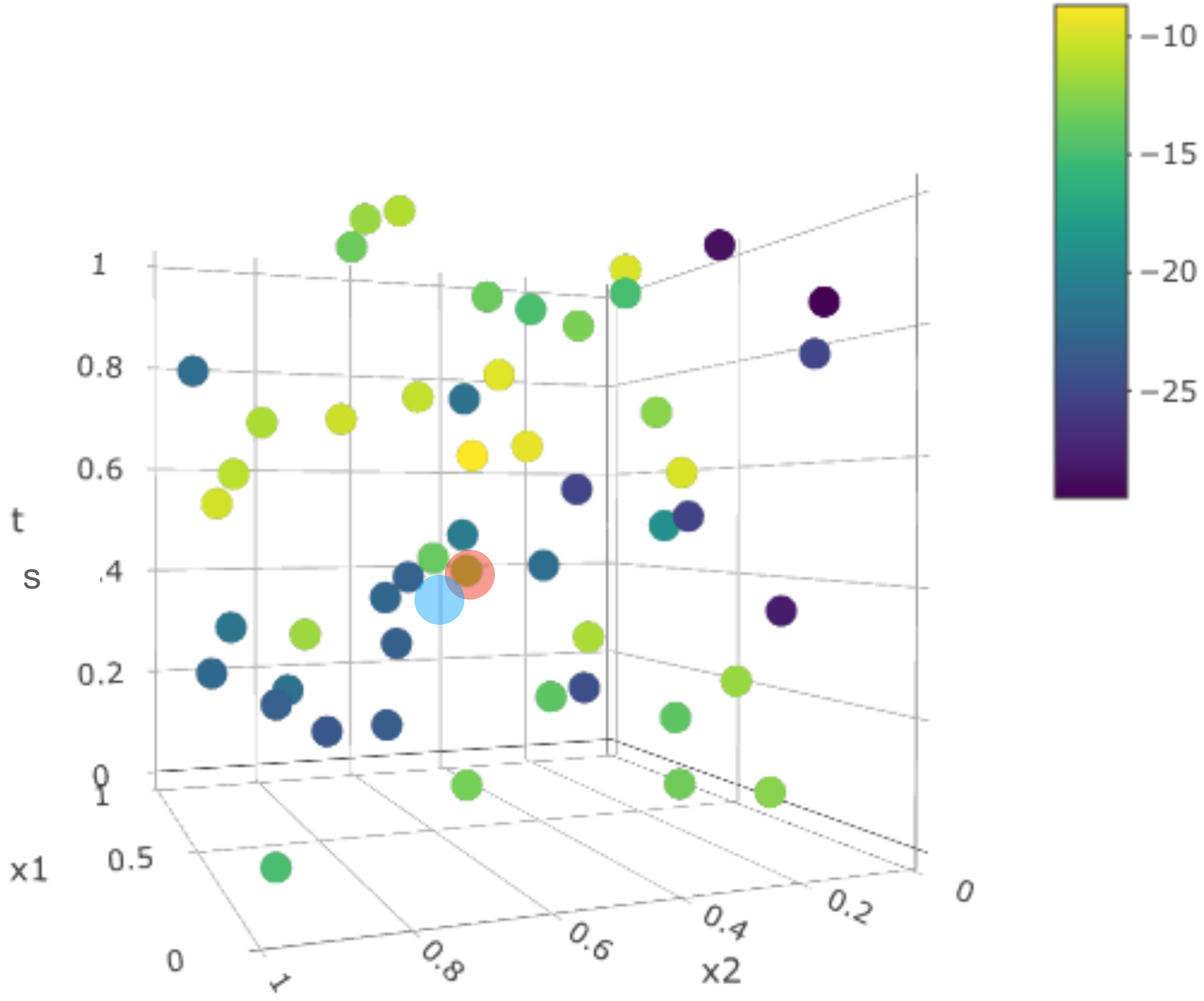




Far in the covariate domain

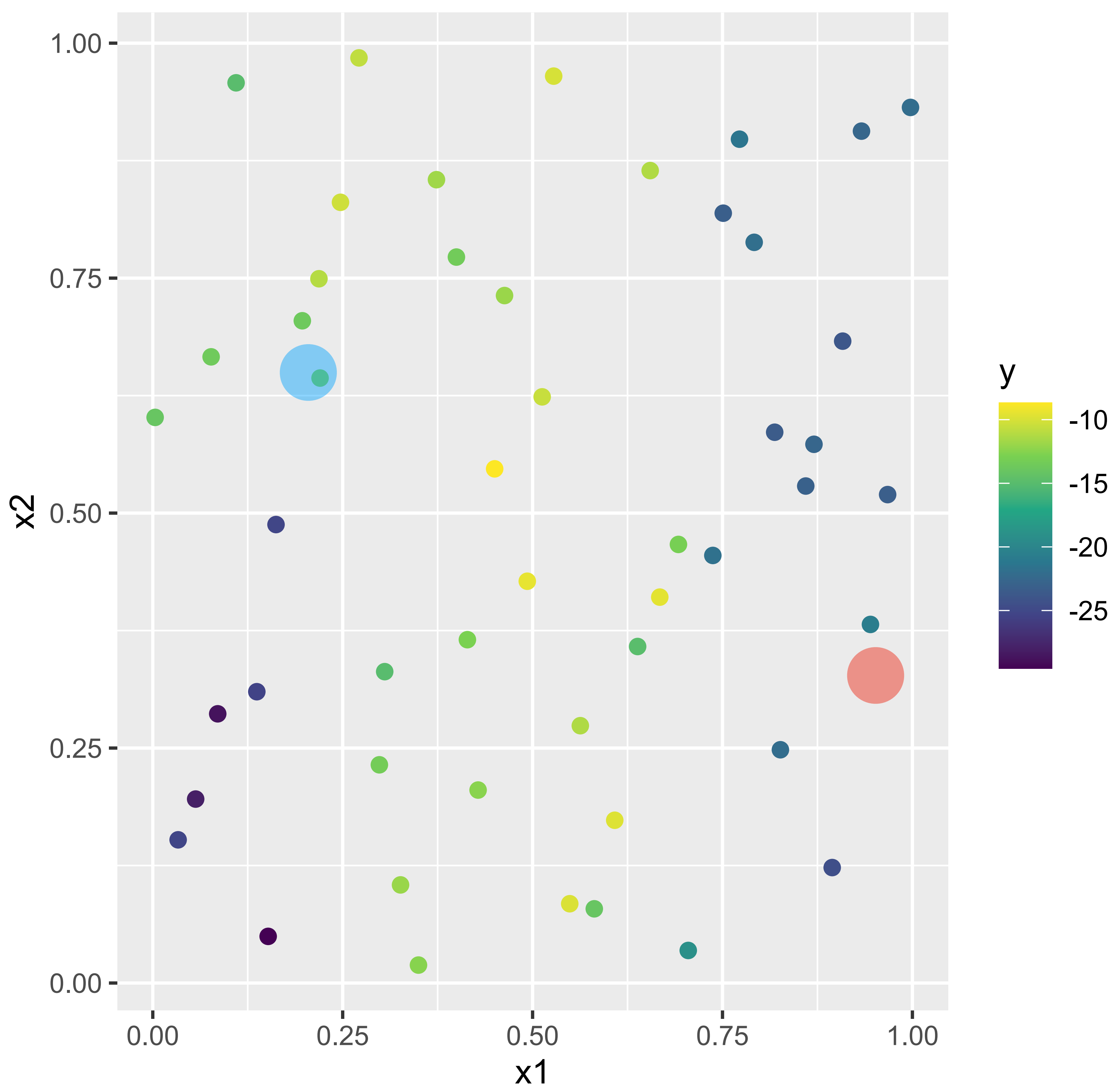
Issues of RF for dependent data

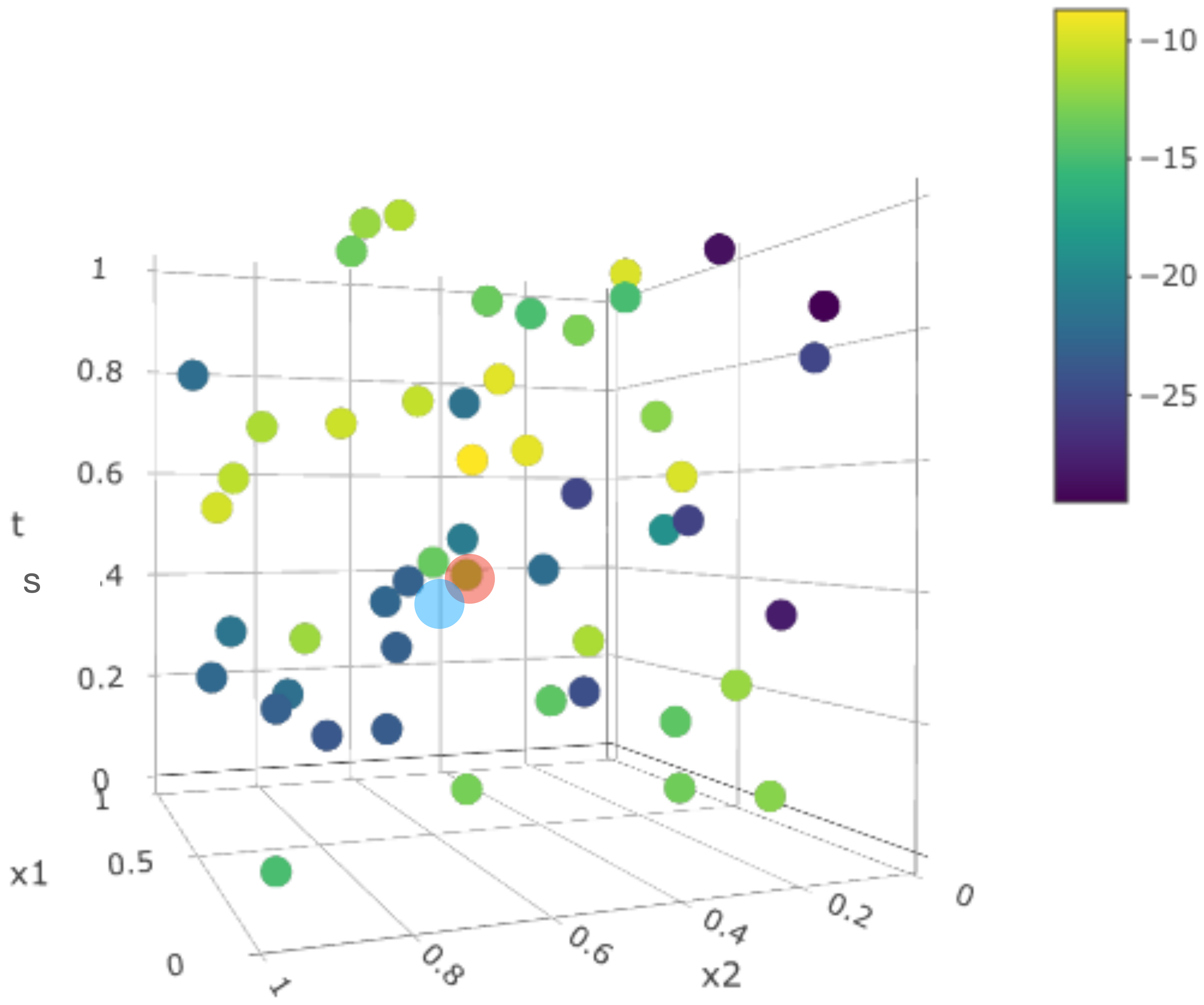




Close in space/time domain and likely to be correlated

Issues of RF for dependent data





Close in space/time domain and likely to be correlated

Local decision making in the regression trees ignores serial/spatial correlation

Issues of RF for dependent data

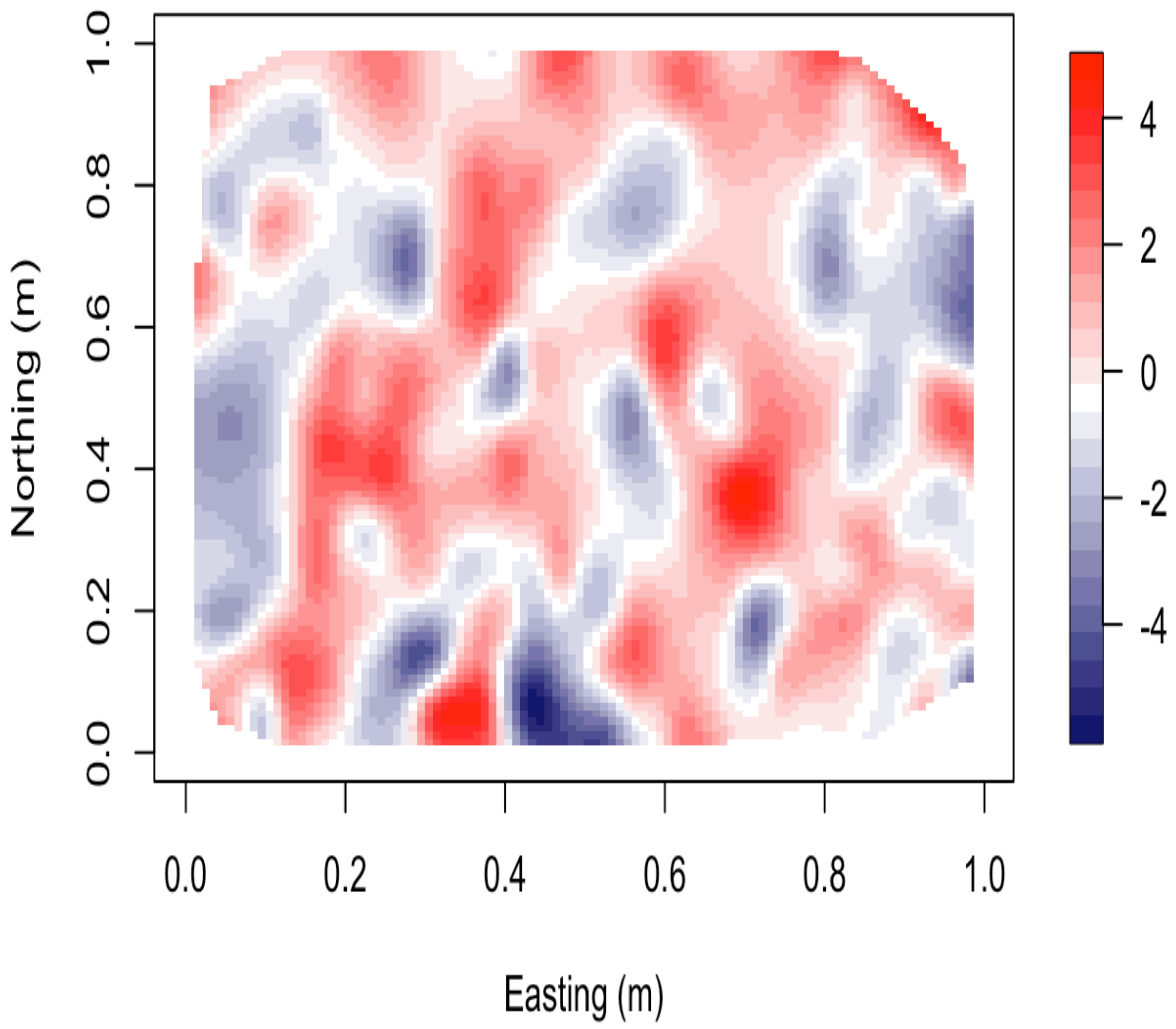
Local decision making in the regression trees ignores correlation with data in other nodes

Use of variances (least squares loss) and node mean as the representative in the CART-split criterion ignores correlation among data within a node

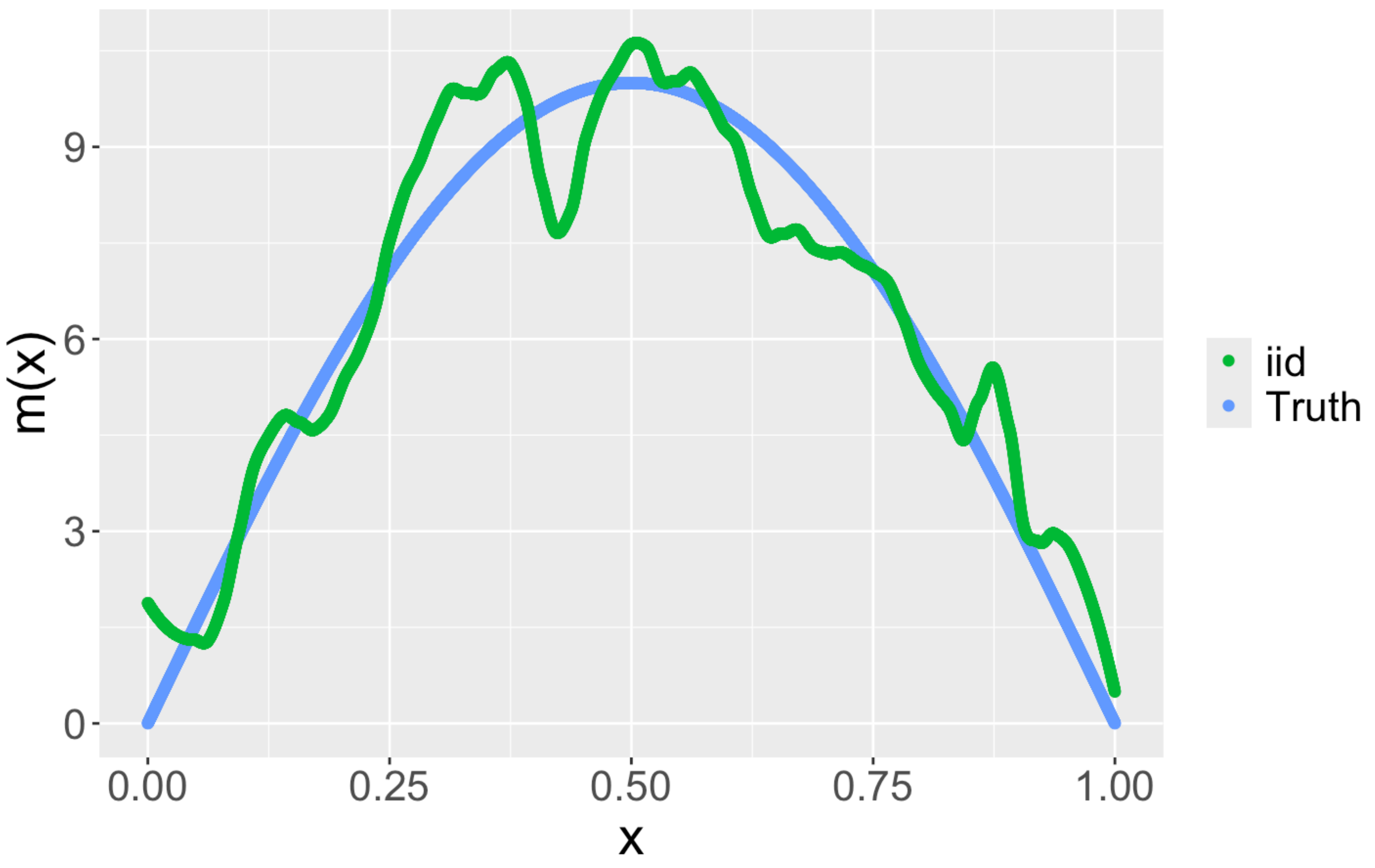
Resampling of data to create a forest of trees is not ideal for correlated data.

Issues of RF for dependent data

$m(x) = 10sin(\pi x)$



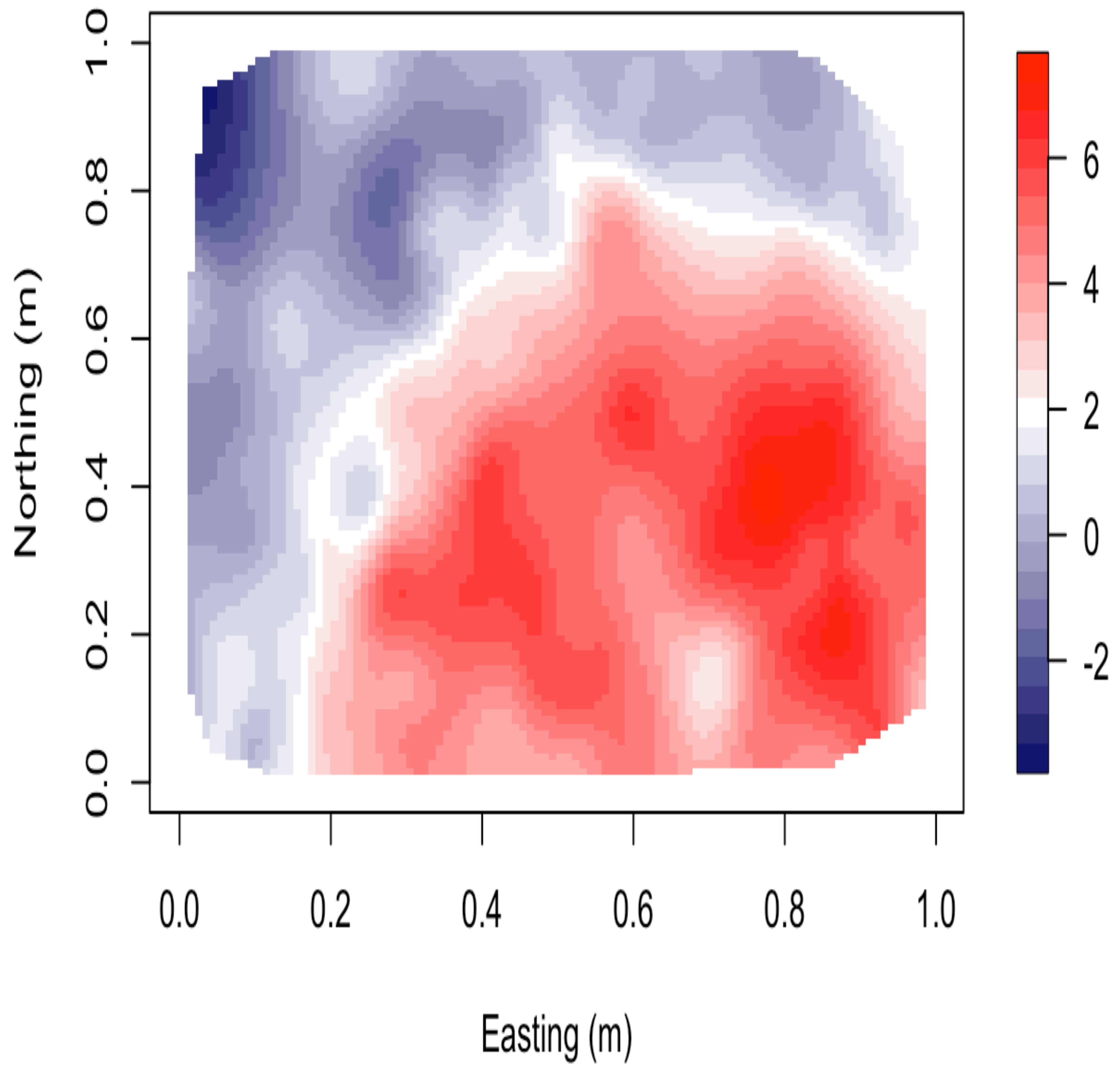
iid errors



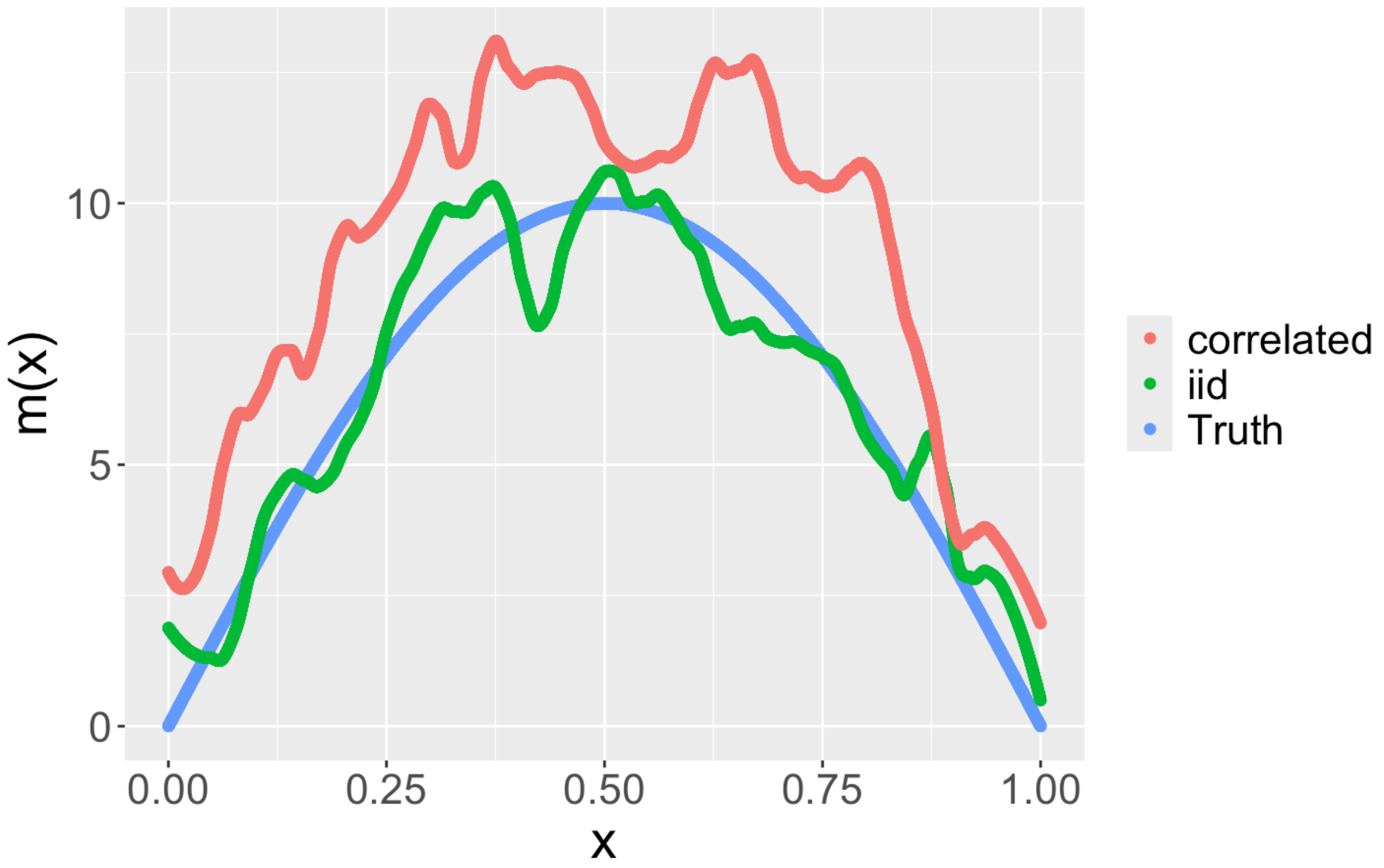
$\widehat{m}(x)$ from RF

Issues of RF for dependent data

$m(x) = 10sin(\pi x)$



spatially correlated errors



$\widehat{m}(x)$ from RF

Random forest methods for geospatial data

1. Naïve approach — Residual/hybrid kriging:

Estimates a non-linear regression function $E(Y) = m(X)$ using Random Forests

Kriging on the residuals $Y_i - \widehat{m}(X_i)$ for spatially-informed predictions

Fayad et al. 2016; Fox et al. 2020

Random forest methods for geospatial data

1. Naïve approach — Residual/hybrid kriging:

Estimates a non-linear regression function $E(Y) = m(X)$ using Random Forests

Kriging on the residuals $Y_i - \widehat{m}(X_i)$ for spatially-informed predictions

Fayad et al. 2016; Fox et al. 2020

Spatial dependence is completely ignored during estimation

- Ignoring spatial correlation impacts estimation
- Poor estimation in turn can affect prediction performance

Random forest methods for geospatial data

2. Brute-force approach — added spatial features

Creates a set of spatial features / covariates $F(s)$

(spatial co-ordinates, pairwise distances, basis functions, etc.)

Estimates a non-linear regression function $E(Y) = g(X, F(s))$ using ML

Random forests: Hengl et al., 2018.

Random forest methods for geospatial data

2. Brute-force approach — added spatial features

Creates a set of spatial features / covariates $F(s)$

(spatial co-ordinates, pairwise distances, basis functions, etc.)

Estimates a non-linear regression function $E(Y) = g(X, F(s))$ using ML

Random forests: Hengl et al., 2018.

Does not belong to the mixed effects model framework

Prediction only! Cannot estimate separate spatial and non-spatial effects

Curse of dimensionality: Often needs a large number of spatial features

Random forest methods for geospatial data

1. Naïve approach — Residual/hybrid kriging
2. Brute-force approach — added spatial features

    Neither approach actually models spatial correlation as is done traditionally in geospatial analysis

Random forest methods for geospatial data

3. Model-based approach — random forests within the spatial mixed model:

$$Y_i = \cancel{X_i^T\beta}\ m(X_i) + w_i + \epsilon_i^*, w \sim GP(0,C), \epsilon_i^* \sim_{iid} N(0,\tau^2)$$

Estimate a non-linear m using Random Forests

Retains all advantages of the traditional spatial mixed models

Interpretability and parsimony of GP

Estimation of mean and spatial prediction (kriging)

Challenge:

How to estimate m using random forests within this model-based framework?

Generalized least squares

$$Y_i = m(X_i) + w_i + \epsilon_i^*, w \sim GP(0,C), \epsilon_i^* \sim_{iid} N(0,\tau^2)$$

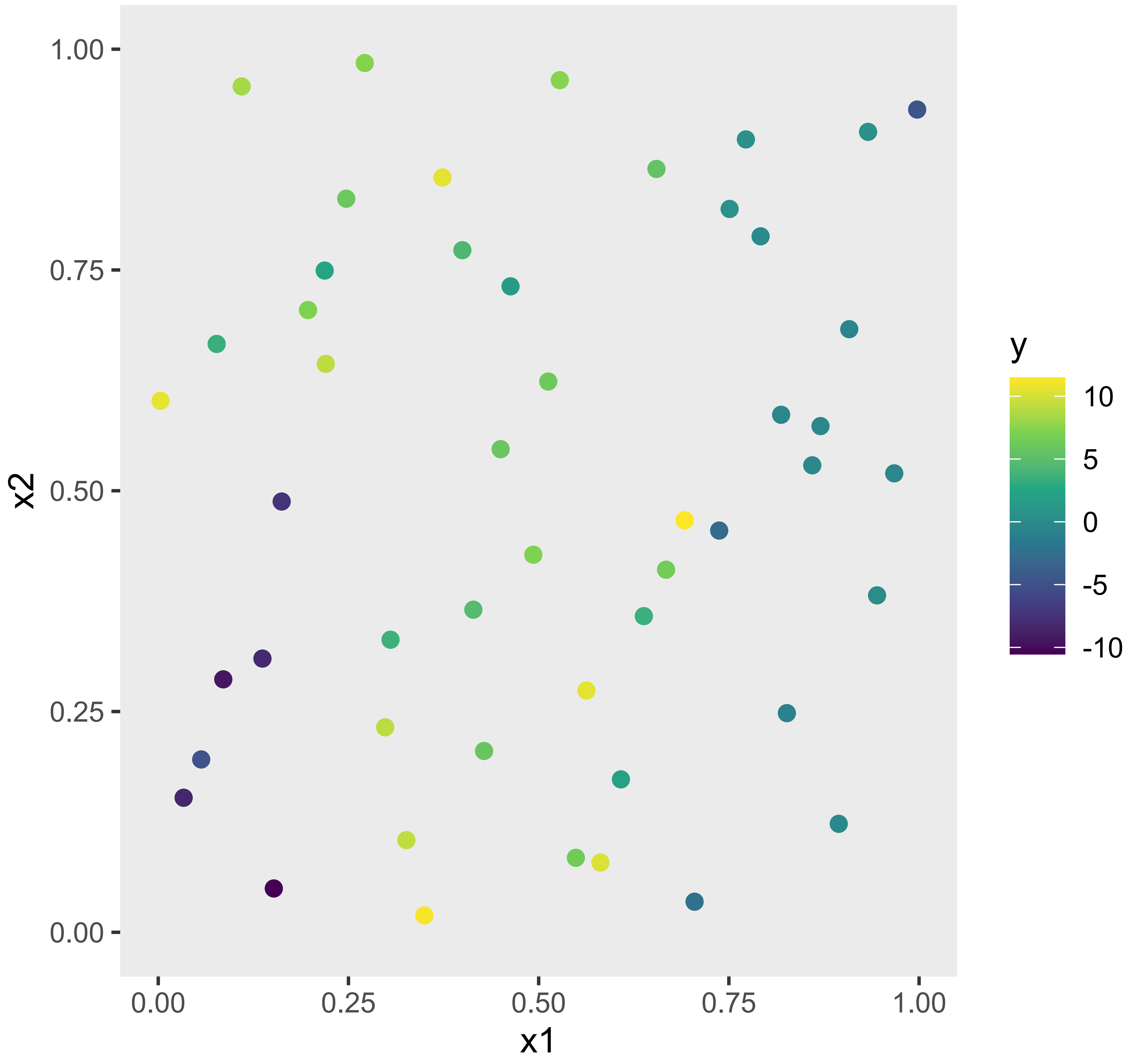
Marginal model: $Y \sim N(m(X), \Sigma)$ where $\Sigma = Cov(\epsilon) = C(\theta) + \tau^2 I$

When $m(X_i) = X_i'\beta$, for a given $\Sigma = Cov(\epsilon)$, the maximum likelihood estimator (MLE) of β is the generalized least squares (GLS) estimate

$$\hat{\beta}_{GLS} = \arg\max_\beta (Y - X\beta)'\Sigma^{-1}(Y - X\beta) = (X'\Sigma^{-1}X)^{-1}X'\Sigma^{-1}Y$$

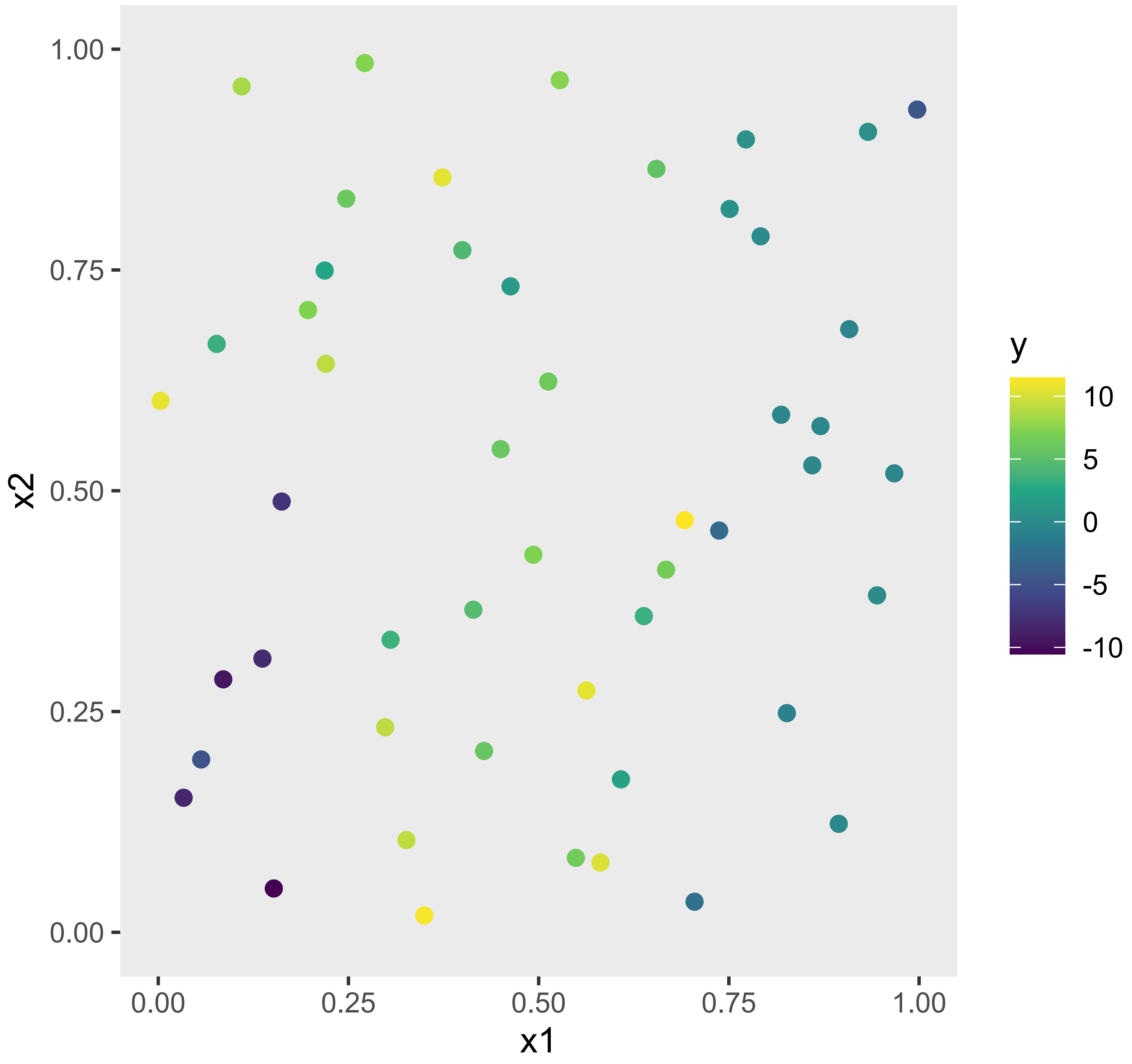
The estimate of the mean function is $\widehat{m(x)} = x'\hat{\beta}_{GLS}$

Revisiting the CART-split criterion



CART (Classification and regression tree) Split criterion: Maximize $\frac{1}{n_P}\left[\sum_{i=1}^{n_P}(Y_i^P-\bar{Y}^P)^2-\sum_{i_r=1}^{n_R}(Y_i^R-\bar{Y}^R)^2-\sum_{i_l=1}^{n_L}(Y_i^L-\bar{Y}^L)^2\right]$

Revisiting the CART-split criterion

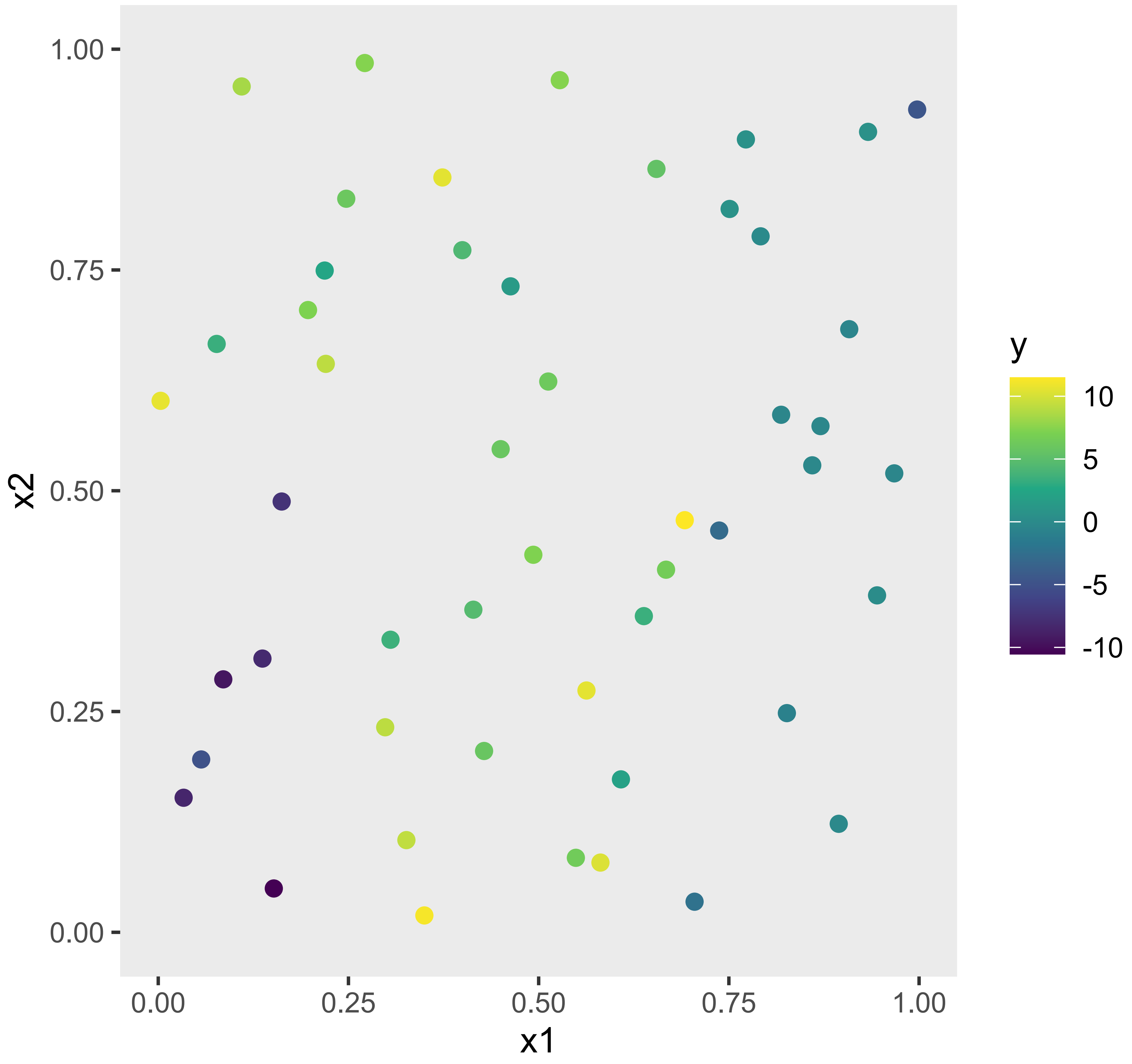


Parent node → $[0,1]^2$

Parent node variance

CART (Classification and regression tree) Split criterion: Maximize $\frac{1}{n_P}\left[\sum_{i=1}^{n_P}(Y_i^P - \bar{Y}^P)^2 - \sum_{i_r=1}^{n_R}(Y_i^R - \bar{Y}^R)^2 - \sum_{i_l=1}^{n_L}(Y_i^L - \bar{Y}^L)^2\right]$

Revisiting the CART-split criterion



Parent node $\boxed{[0,1]^2}$

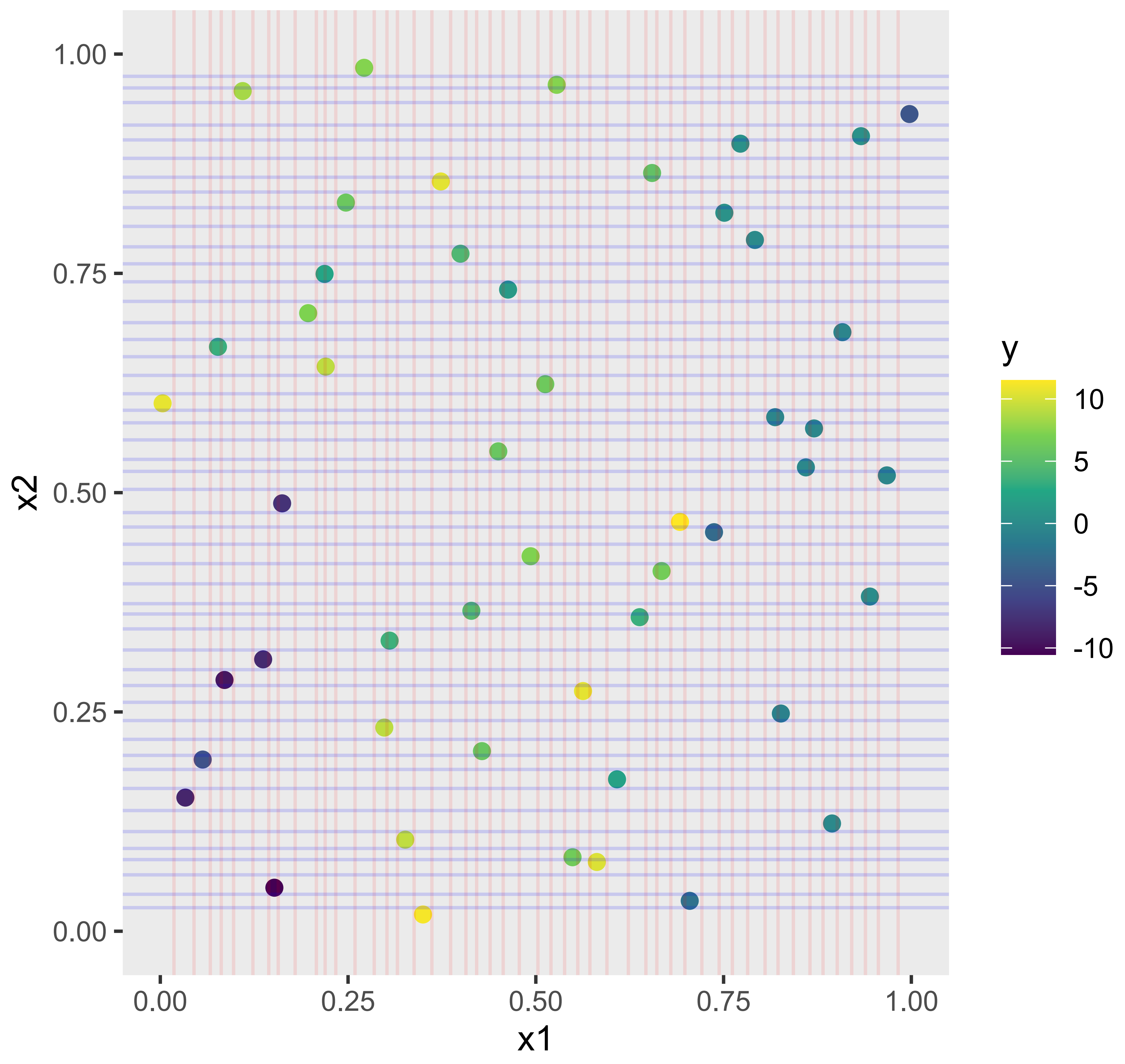
Parent node variance

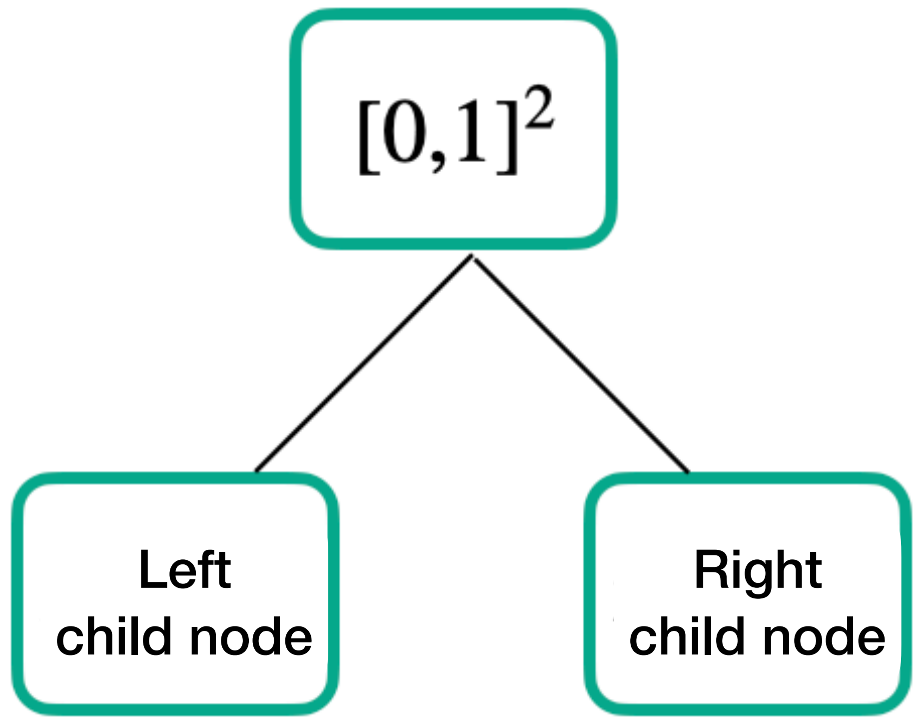
$$\underbrace{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{n-2} \\ y_{n-1} \\ y_n \end{bmatrix}}_{Y} = \underbrace{\begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \\ 1 \\ 1 \end{bmatrix}}_{Z^{(0)}} \beta^{(0)} + \varepsilon$$

$$\min_{\beta^{(0)}} \|Y - Z^{(0)}\beta^{(0)}\|^2$$

CART (Classification and regression tree) Split criterion: Maximize $\frac{1}{n_P}\left[\sum_{i=1}^{n_P}(Y_i^P - \bar{Y}^P)^2 - \sum_{i_r=1}^{n_R}(Y_i^R - \bar{Y}^R)^2 - \sum_{i_l=1}^{n_L}(Y_i^L - \bar{Y}^L)^2\right]$

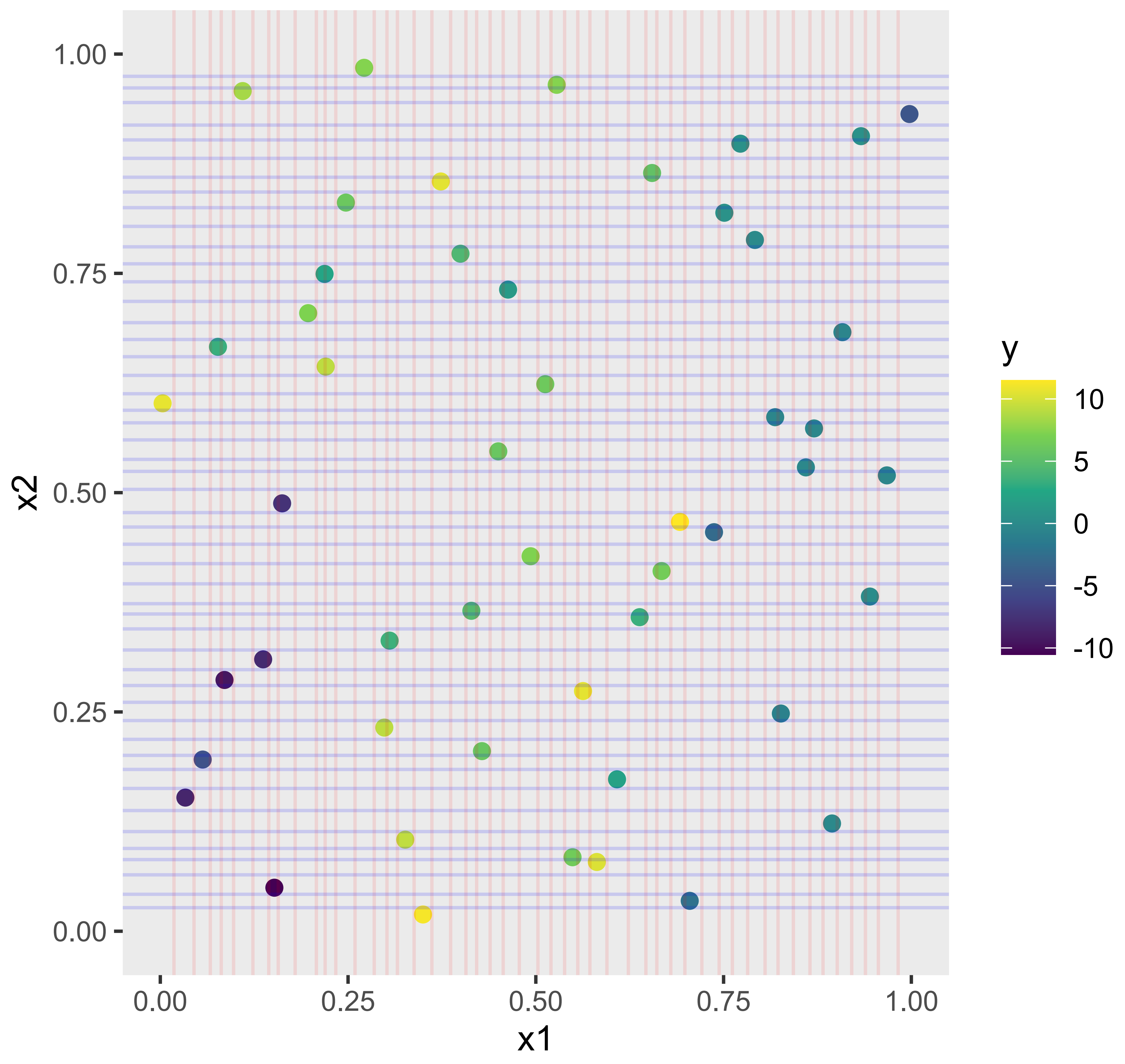
Revisiting the CART-split criterion

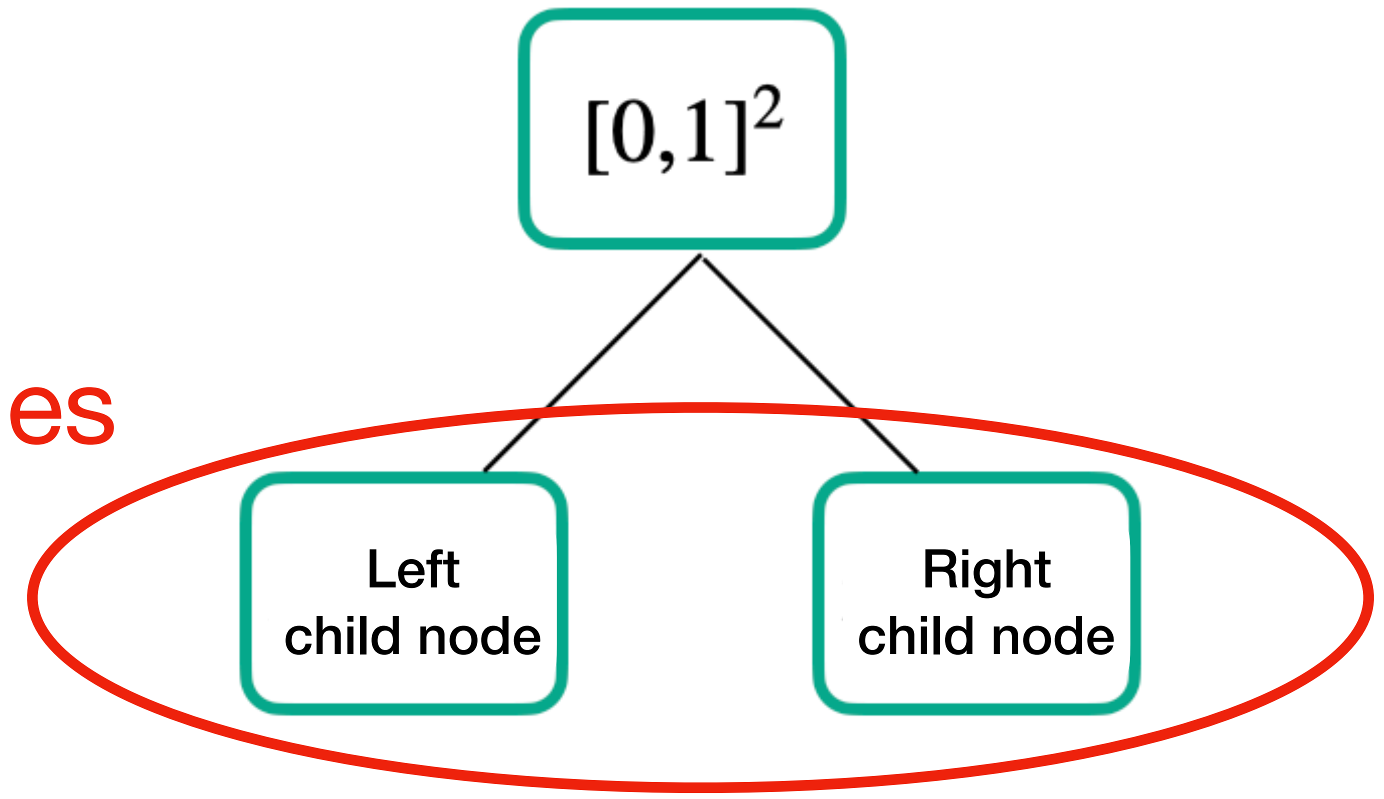




CART (Classification and regression tree) Split criterion: Maximize $\frac{1}{n_P}\left[\sum_{i=1}^{n_P}(Y_i^P - \bar{Y}^P)^2 - \sum_{i_r=1}^{n_R}(Y_i^R - \bar{Y}^R)^2 - \sum_{i_l=1}^{n_L}(Y_i^L - \bar{Y}^L)^2\right]$

Revisiting the CART-split criterion



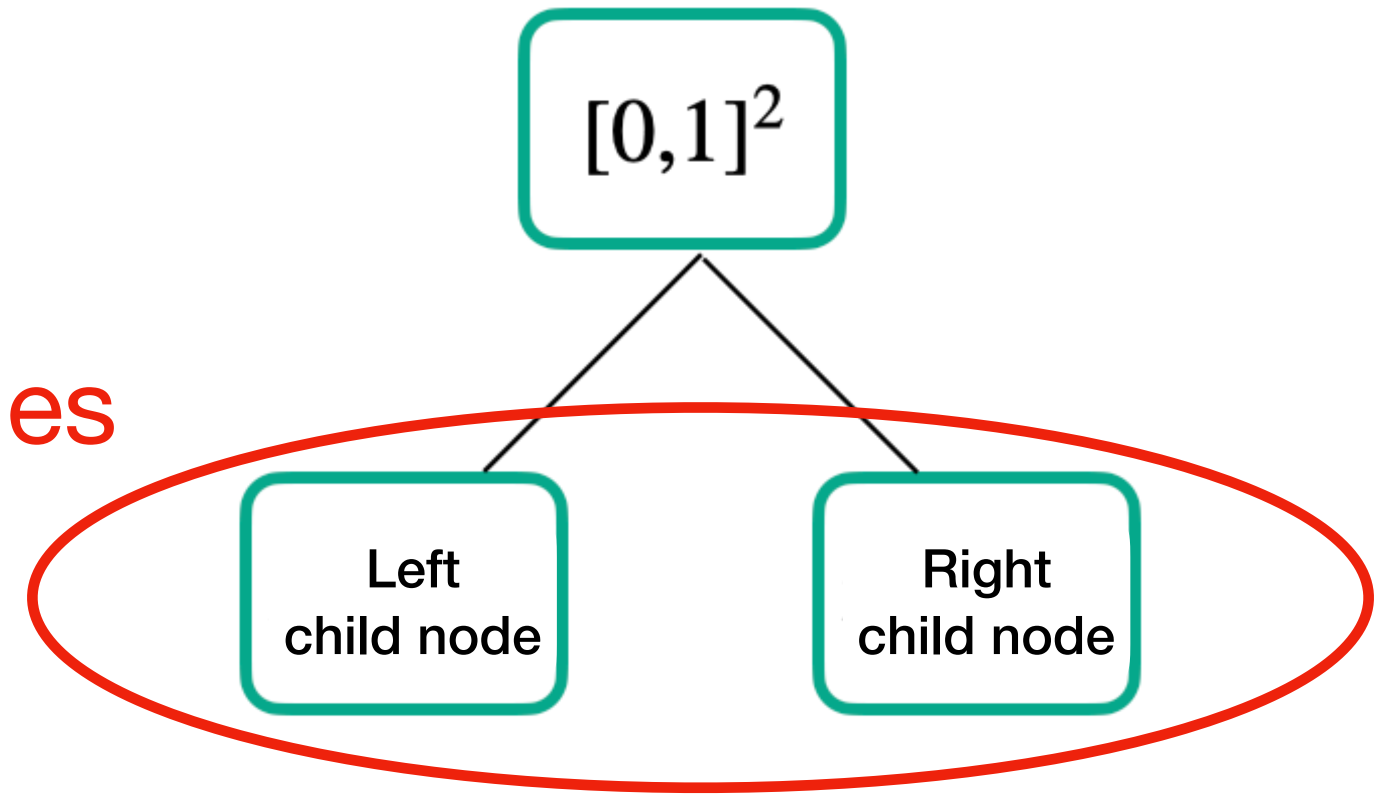


Potential children nodes

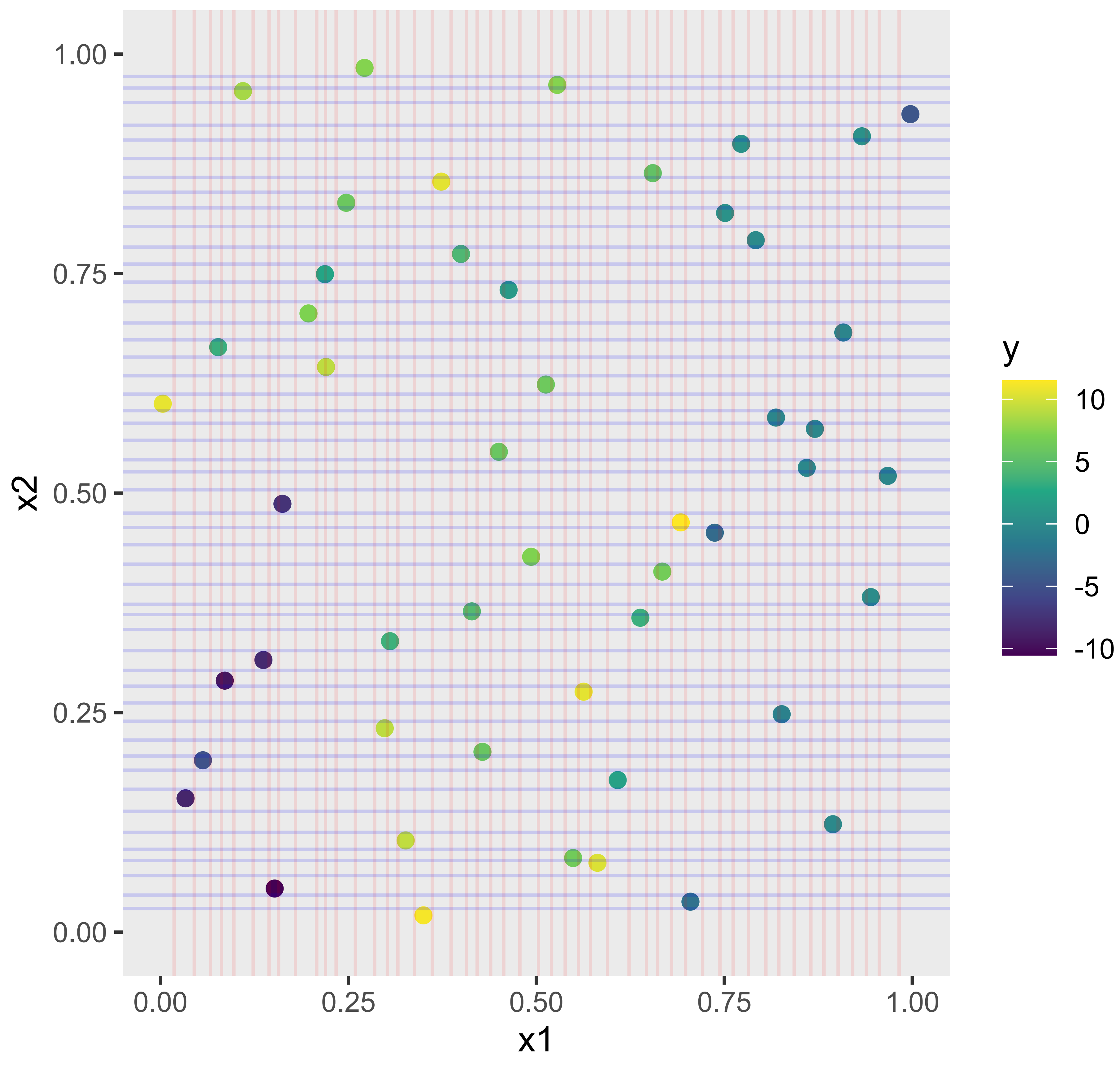
Total children node variance

CART (Classification and regression tree) Split criterion: Maximize $\frac{1}{n_P}\left[\sum_{i=1}^{n_P}(Y_i^P - \bar{Y}^P)^2 - \sum_{i_r=1}^{n_R}(Y_i^R - \bar{Y}^R)^2 - \sum_{i_l=1}^{n_L}(Y_i^L - \bar{Y}^L)^2\right]$

Revisiting the CART-split criterion



Potential children nodes



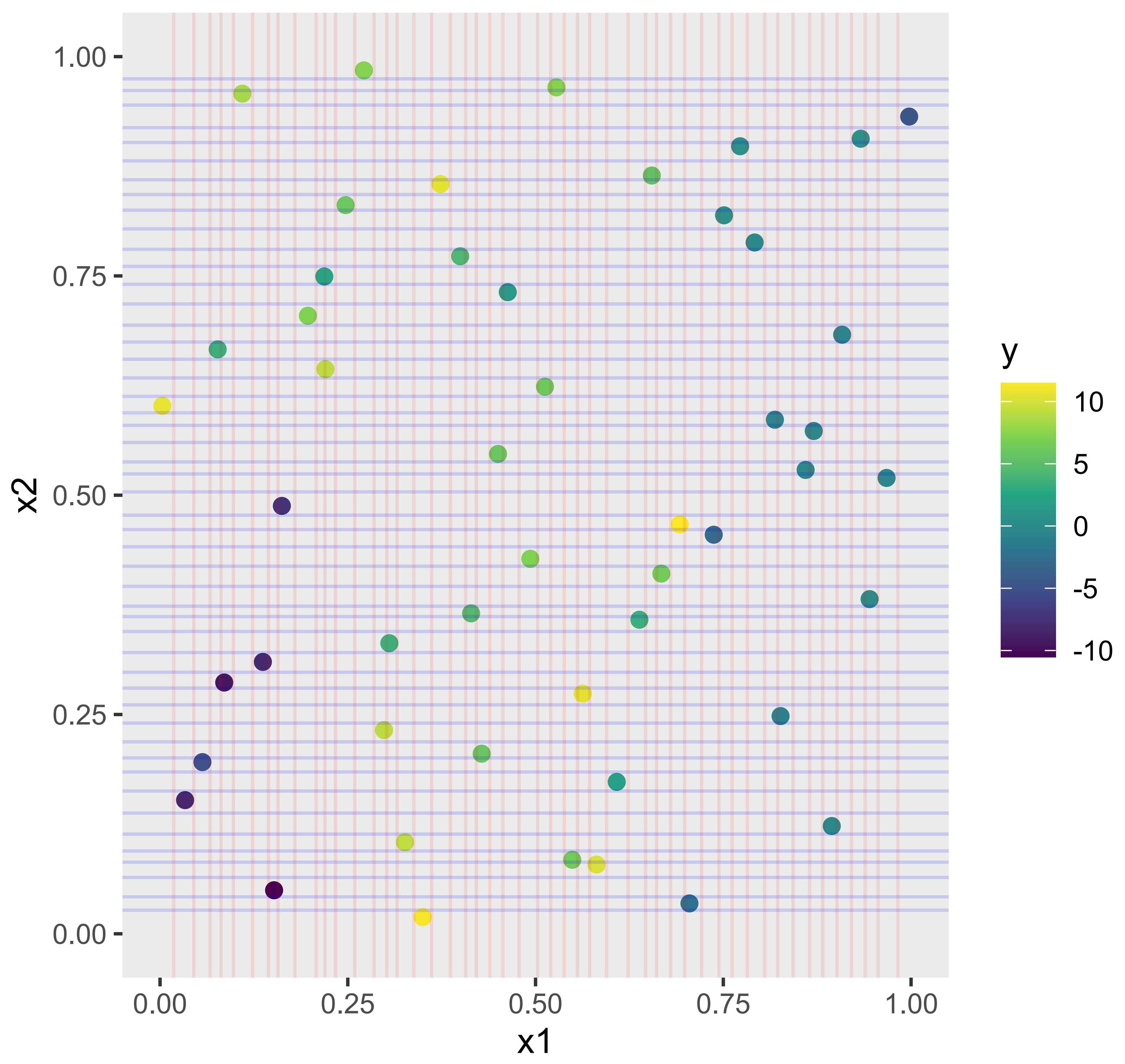
$$\underbrace{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{n-2} \\ y_{n-1} \\ y_n \end{bmatrix}}_{Y} = \underbrace{\begin{array}{cc} \text{Left node membership} & \text{Right node membership} \\ \left[\begin{array}{c} 1 \\ 1 \\ 1 \\ \vdots \\ 0 \\ 0 \\ 0 \end{array}\right. & \left.\begin{array}{c} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \\ 1 \\ 1 \end{array}\right] \end{array}}_{Z} \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} + \varepsilon$$

Total children node variance

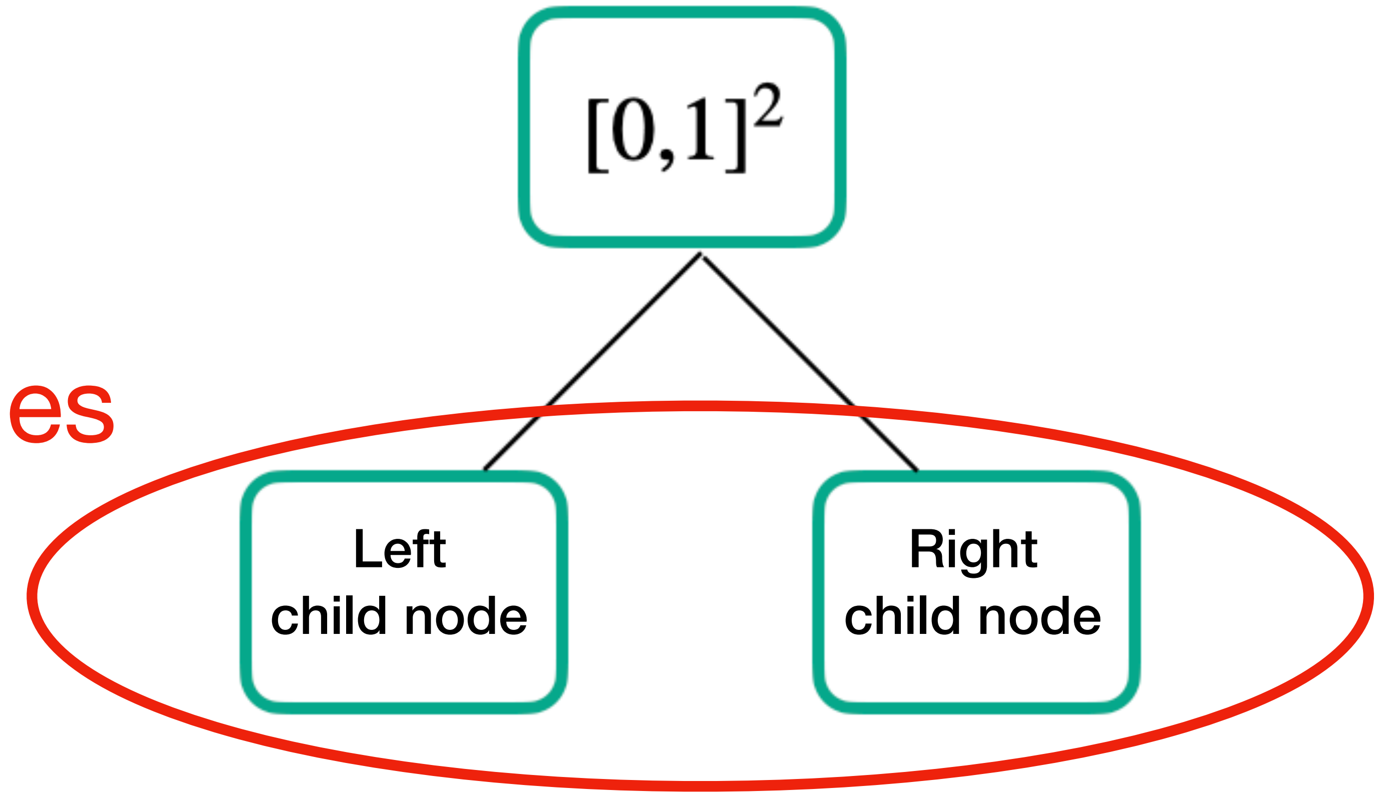
$$\min_{\beta} \|Y - Z\beta\|^2$$

CART (Classification and regression tree) Split criterion: Maximize $\frac{1}{n_P}\left[\sum_{i=1}^{n_P}(Y_i^P - \bar{Y}^P)^2 - \sum_{i_r=1}^{n_R}(Y_i^R - \bar{Y}^R)^2 - \sum_{i_l=1}^{n_L}(Y_i^L - \bar{Y}^L)^2\right]$

Revisiting the CART-split criterion



Potential children nodes



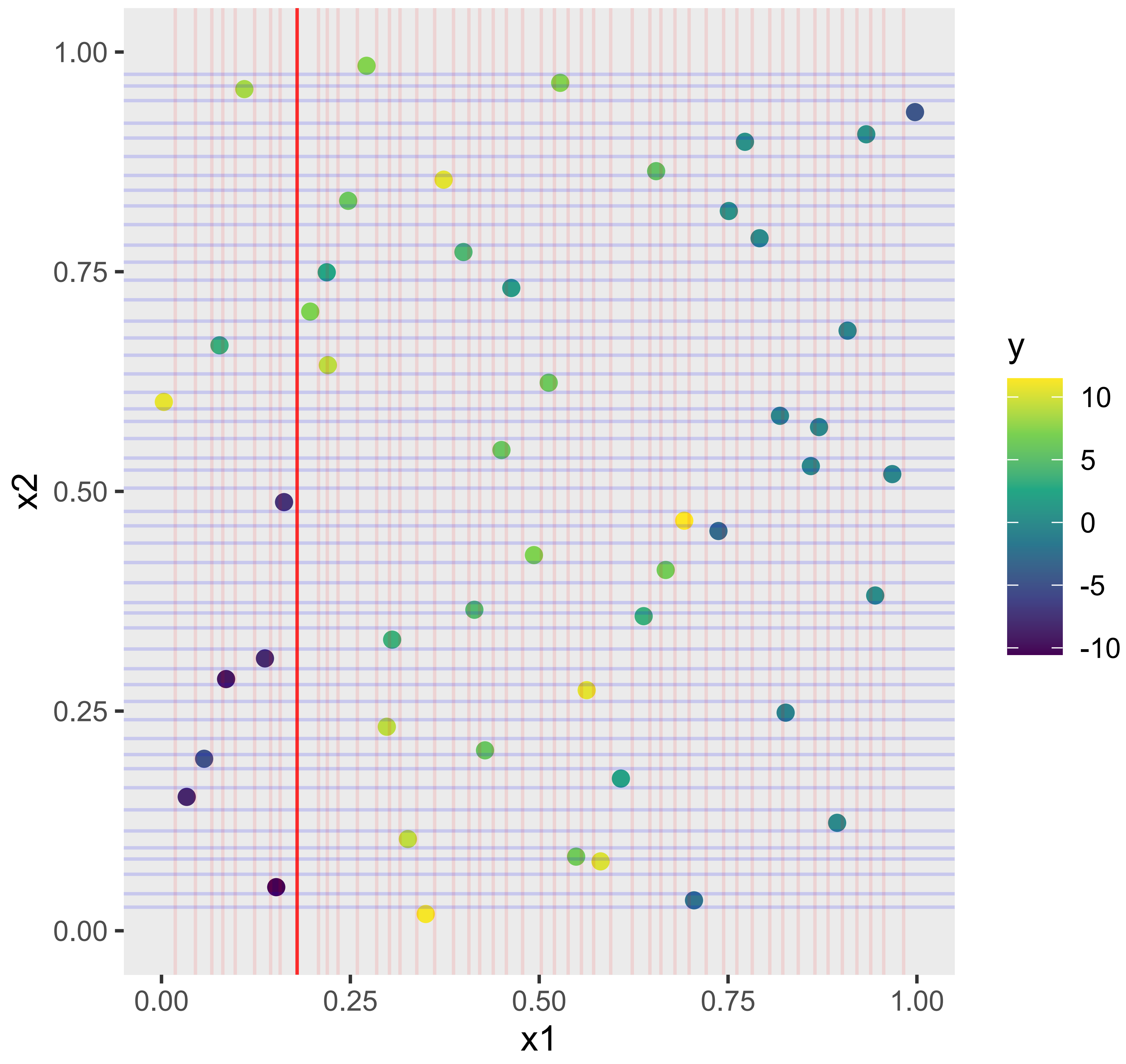
Total children node variance

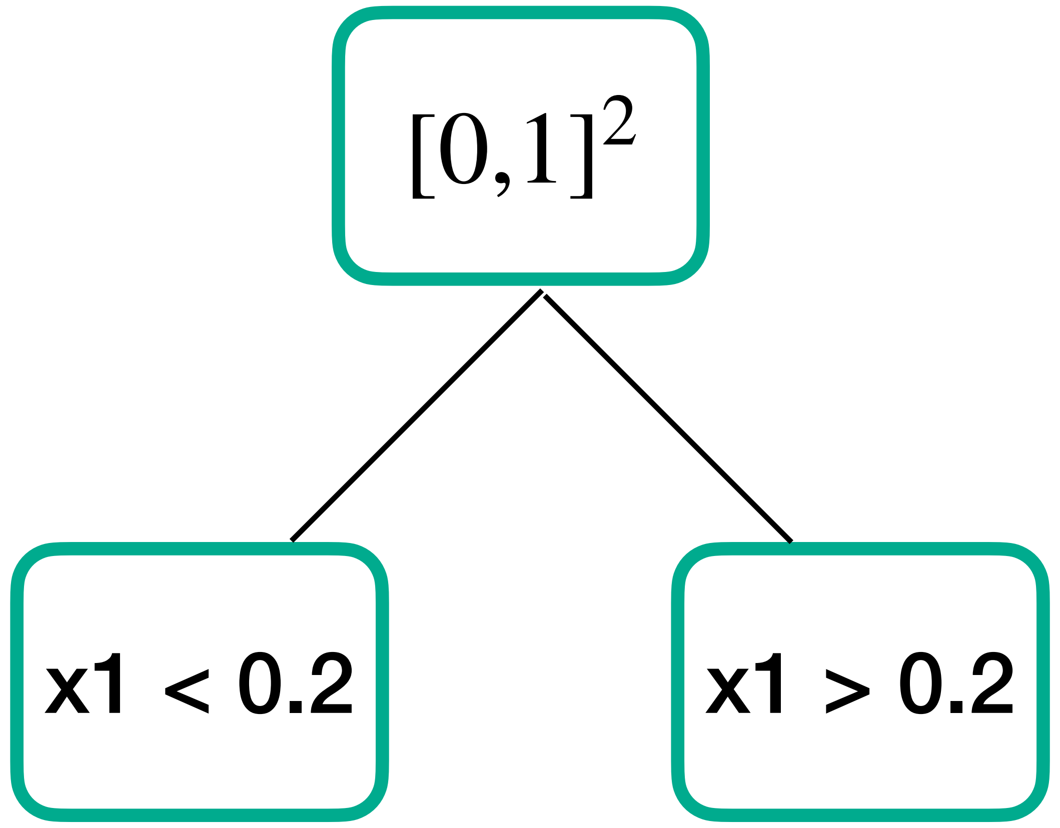
$$\underbrace{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{n-2} \\ y_{n-1} \\ y_n \end{bmatrix}}_{Y} = \begin{array}{cc} \text{Left node membership} & \text{Right node membership} \\ \end{array} \underbrace{\begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ \vdots & \vdots \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}}_{Z} \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} + \varepsilon$$

$$\min_{\beta} \|Y - Z\beta\|^2$$

CART (Classification and regression tree) Split criterion: $\max_{Z,\beta} \frac{1}{n}\left(\|y - Z^{(0)}\hat{\beta}^{(0)}\|_2^2 - \|y - Z\beta\|_2^2 \right)$

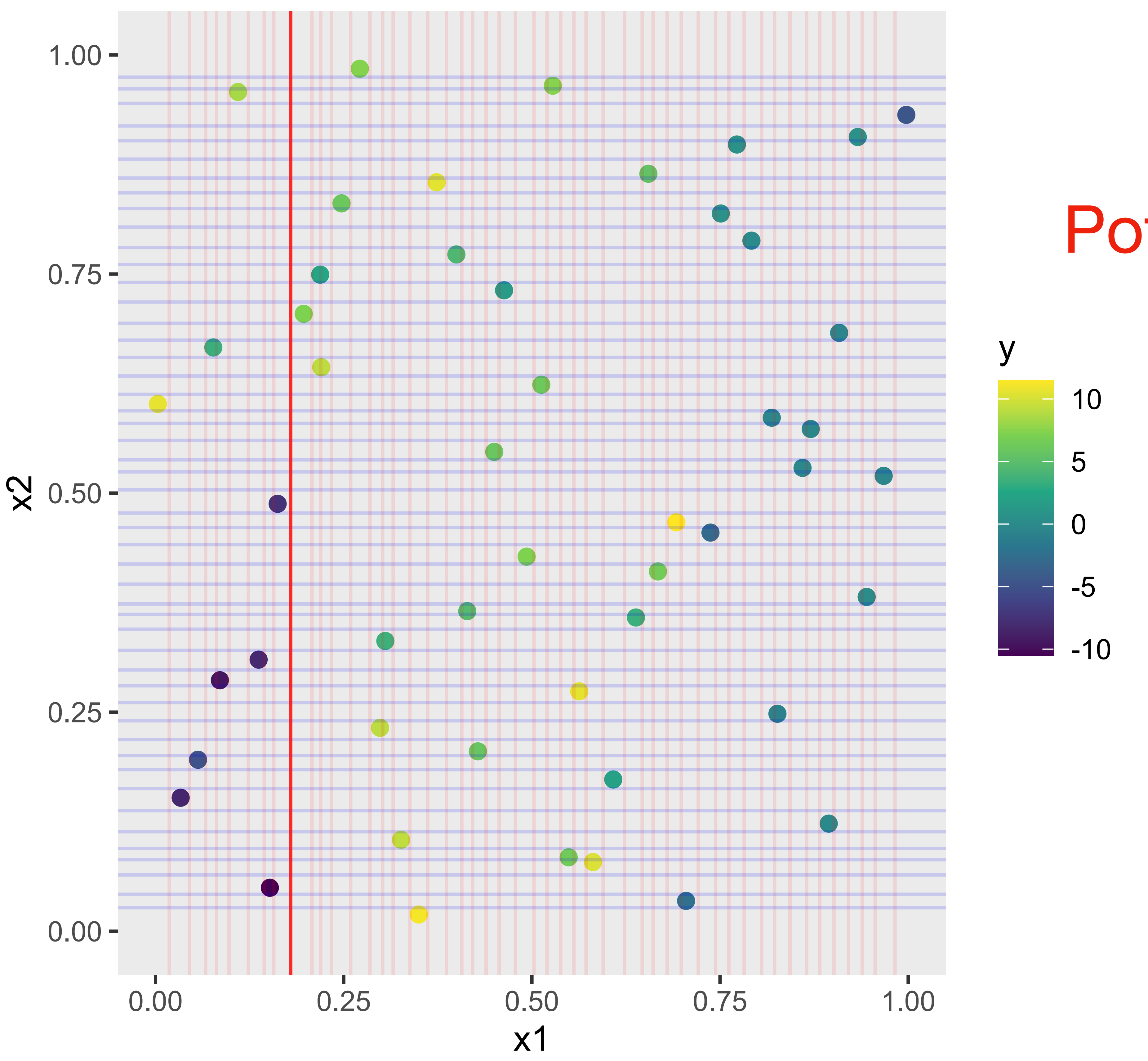
Revisiting the CART-split criterion

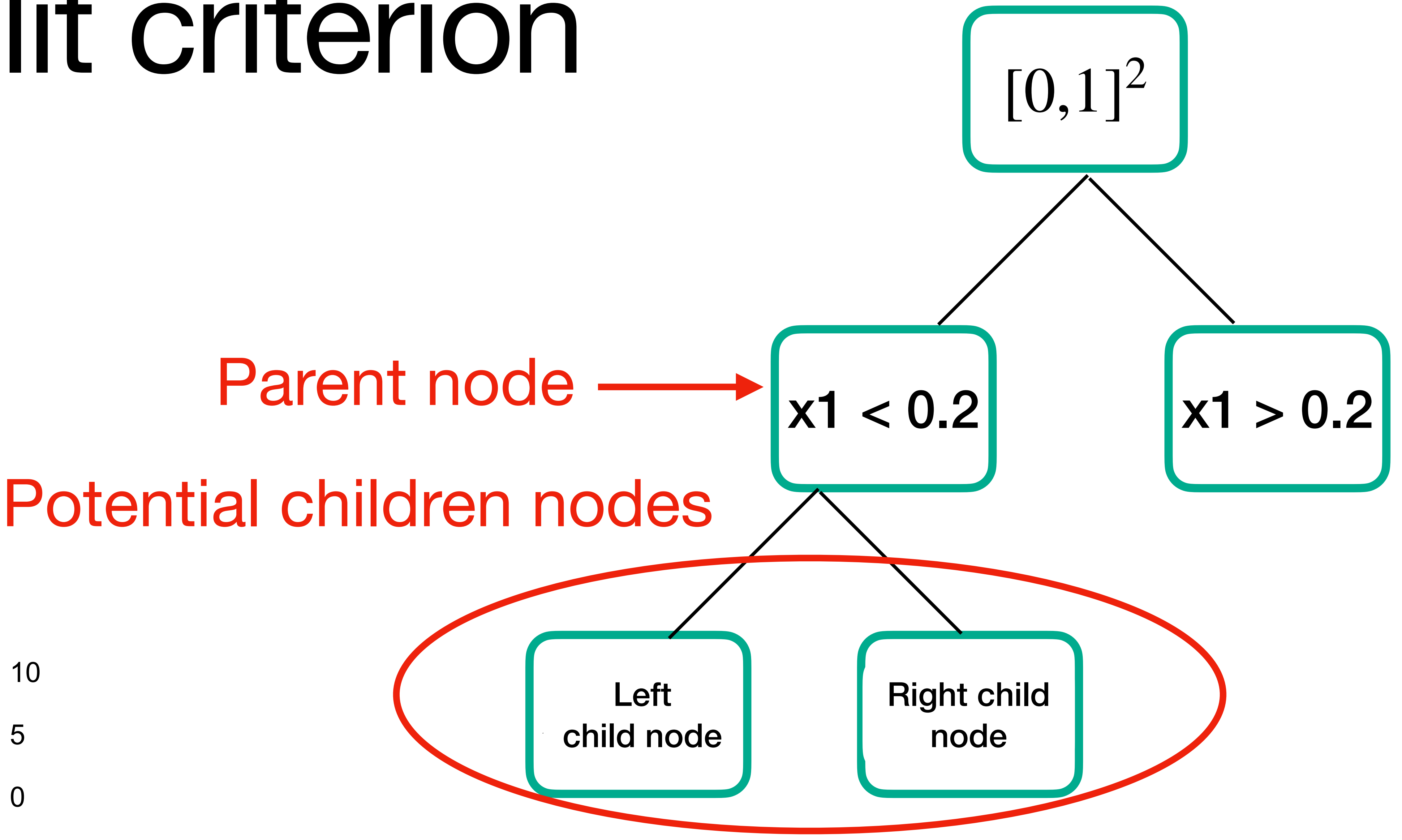




CART (Classification and regression tree) Split criterion: $\max_{Z,\beta} \frac{1}{n}\left(\|y - Z^{(0)}\hat{\beta}^{(0)}\|_2^2 - \|y - Z\beta\|_2^2\right)$

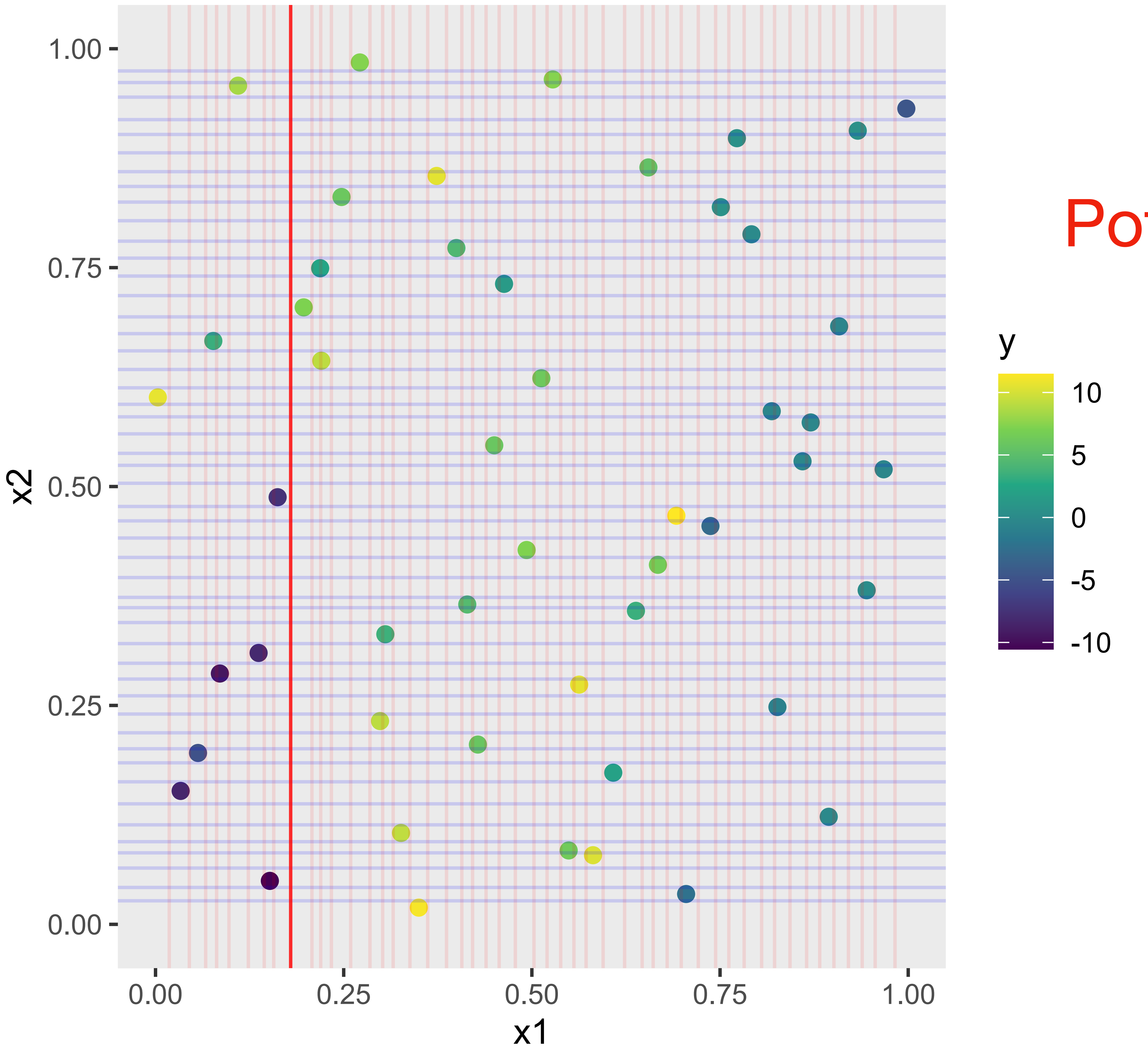
Revisiting the CART-split criterion

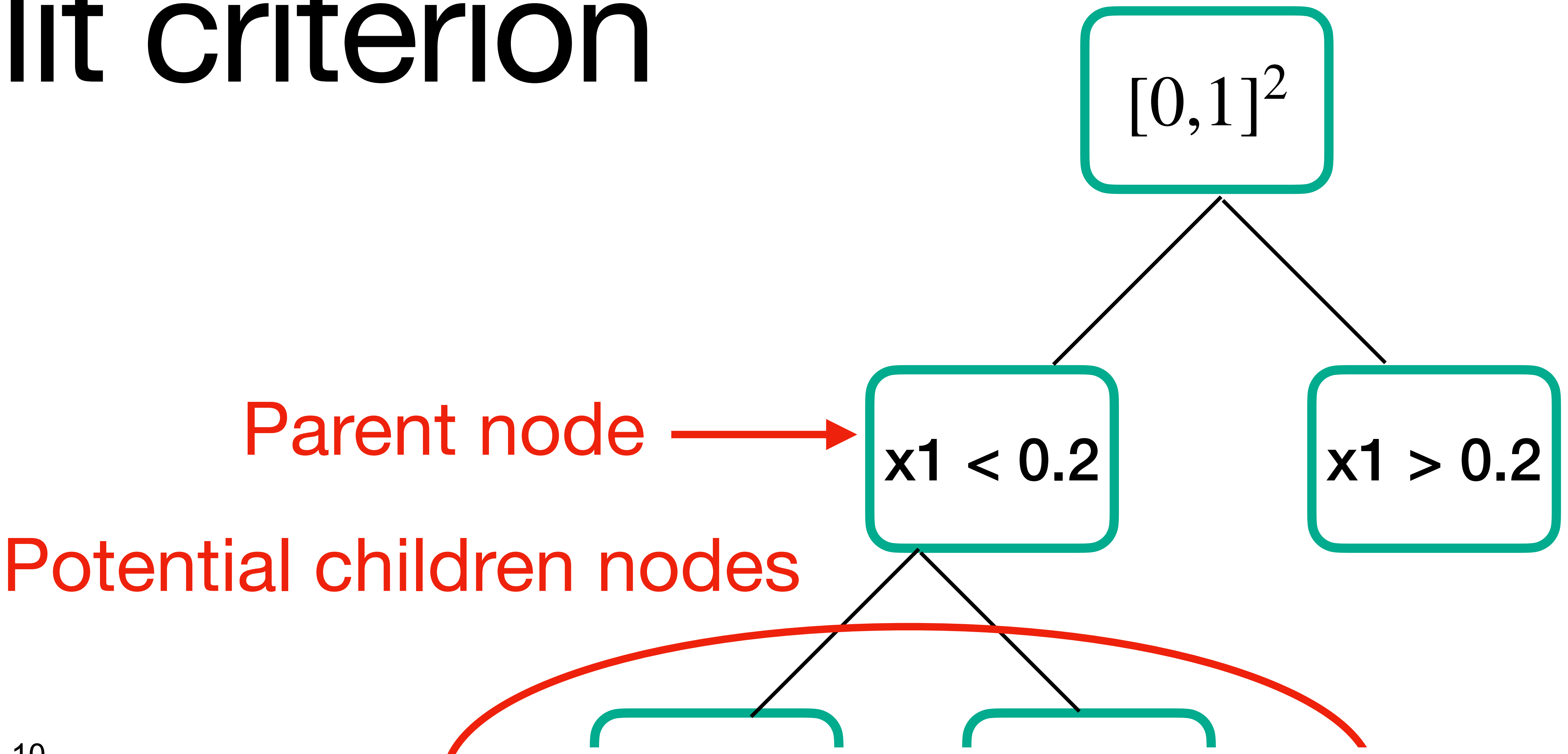




CART (Classification and regression tree) Split criterion: $\max_{Z,\beta} \frac{1}{n}\left(\|y - Z^{(0)}\hat{\beta}^{(0)}\|_2^2 - \|y - Z\beta\|_2^2\right)$

Revisiting the CART-split criterion



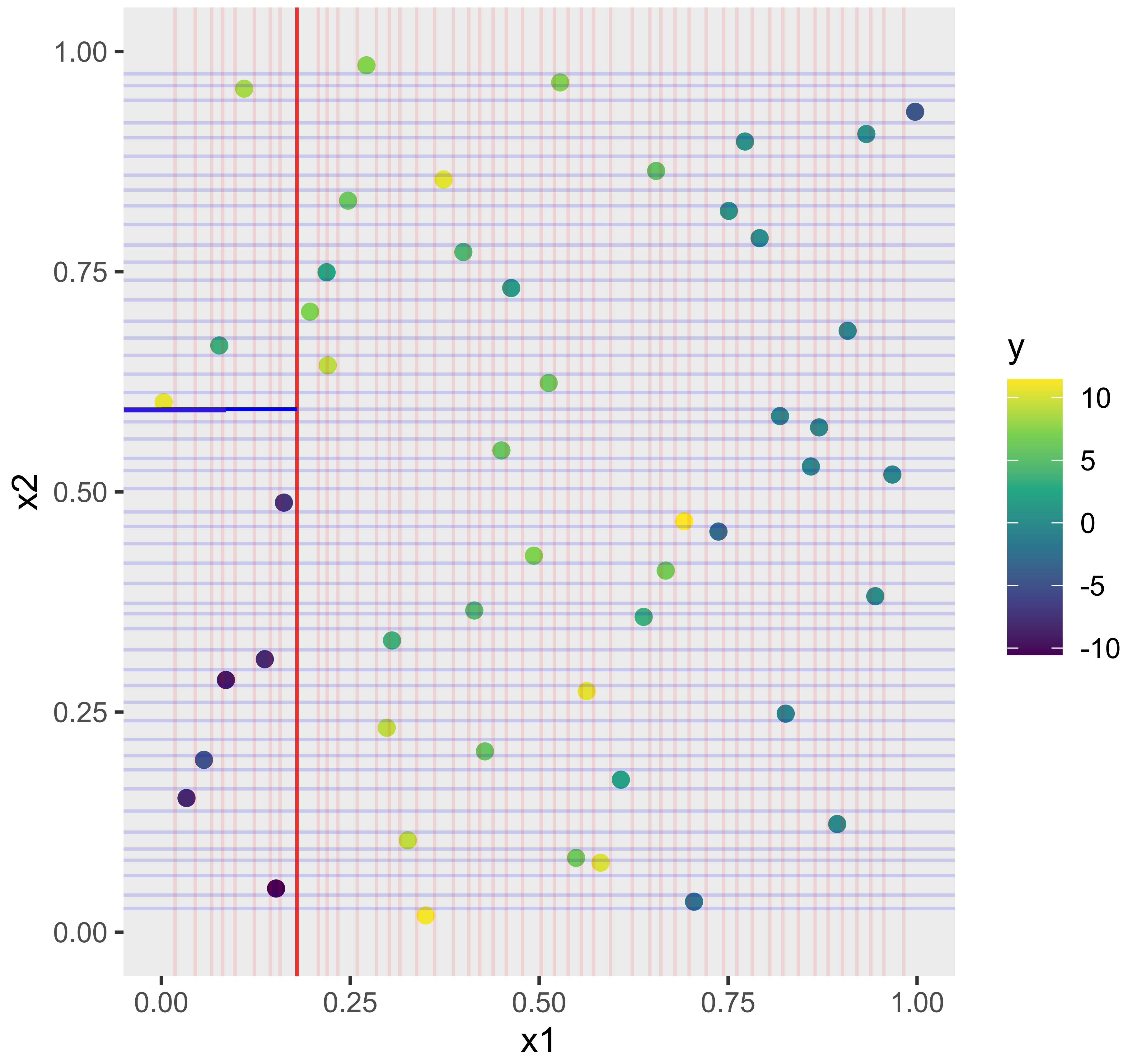


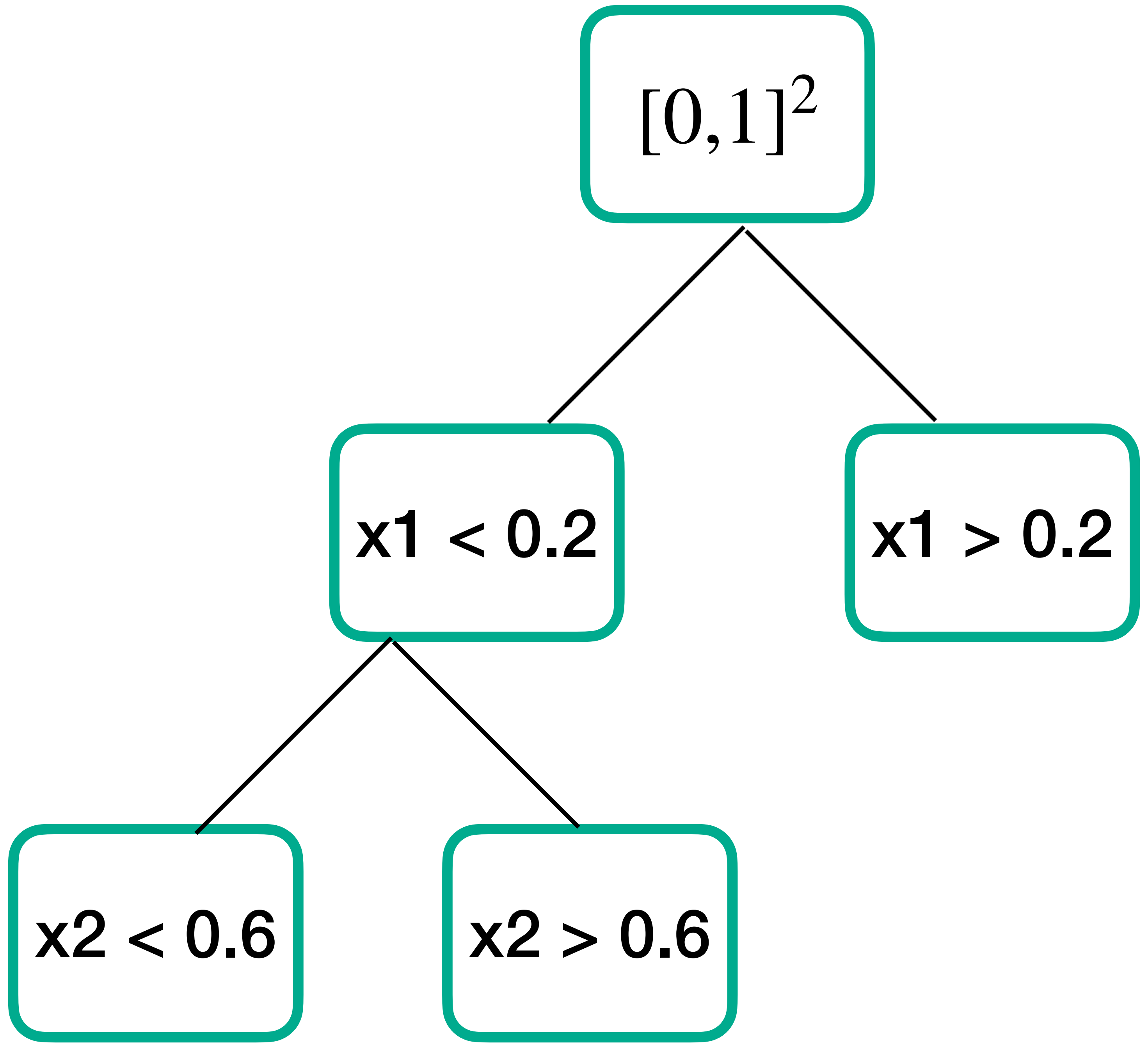
$$\underbrace{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{n-2} \\ y_{n-1} \\ y_n \end{bmatrix}}_{Y} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ \vdots & \vdots & \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}}_{Z} \begin{bmatrix} \beta_{11} \\ \beta_{12} \\ \beta_2 \end{bmatrix} + \varepsilon$$

CART (Classification and regression tree) Split criterion: $\max_{Z,\beta} \frac{1}{n}\left(\|y - Z^{(0)}\hat{\beta}^{(0)}\|_2^2 - \|y - Z\beta\|_2^2 \right)$

Review of Regression Tress and Random Forests

Regression trees

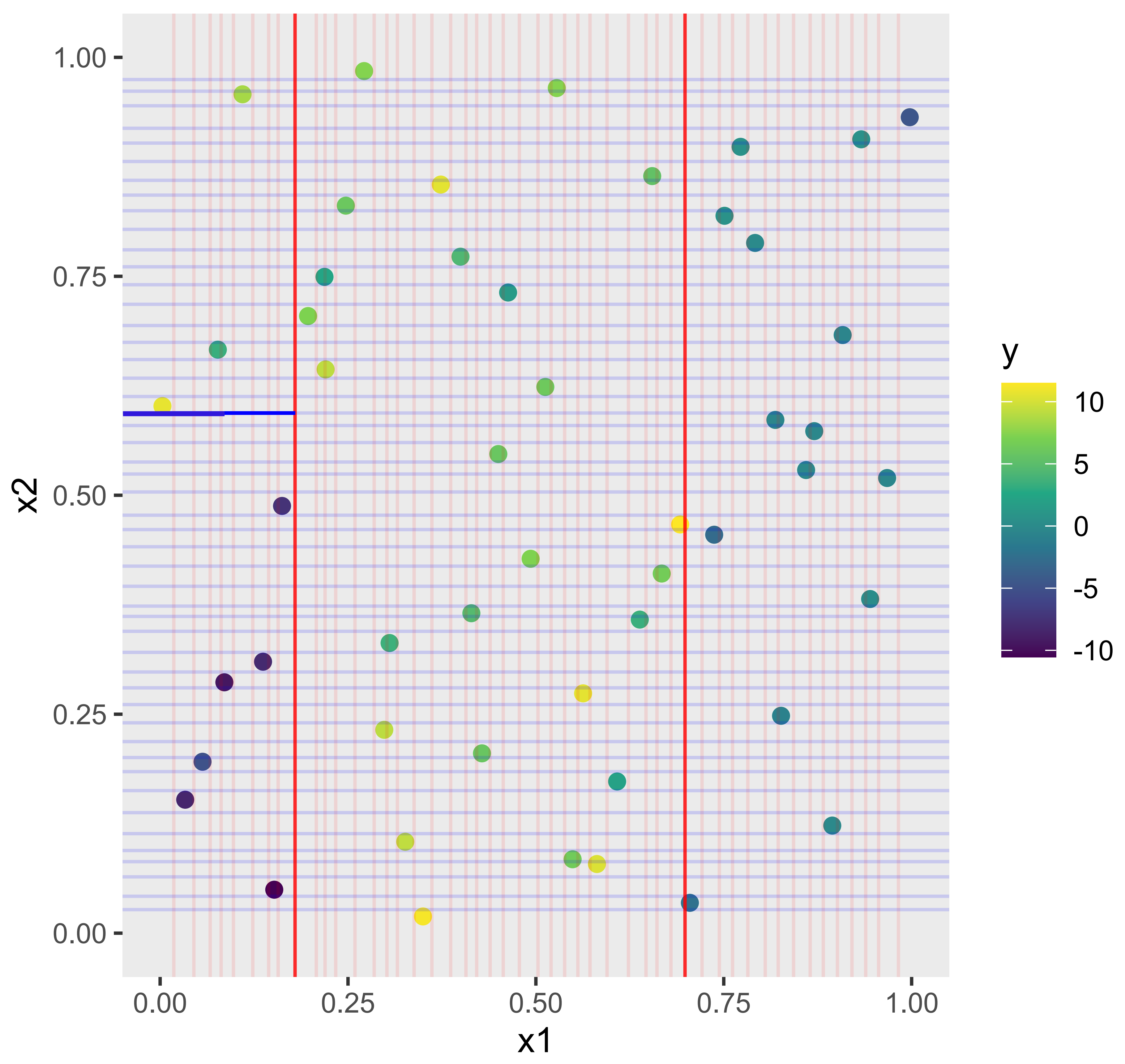


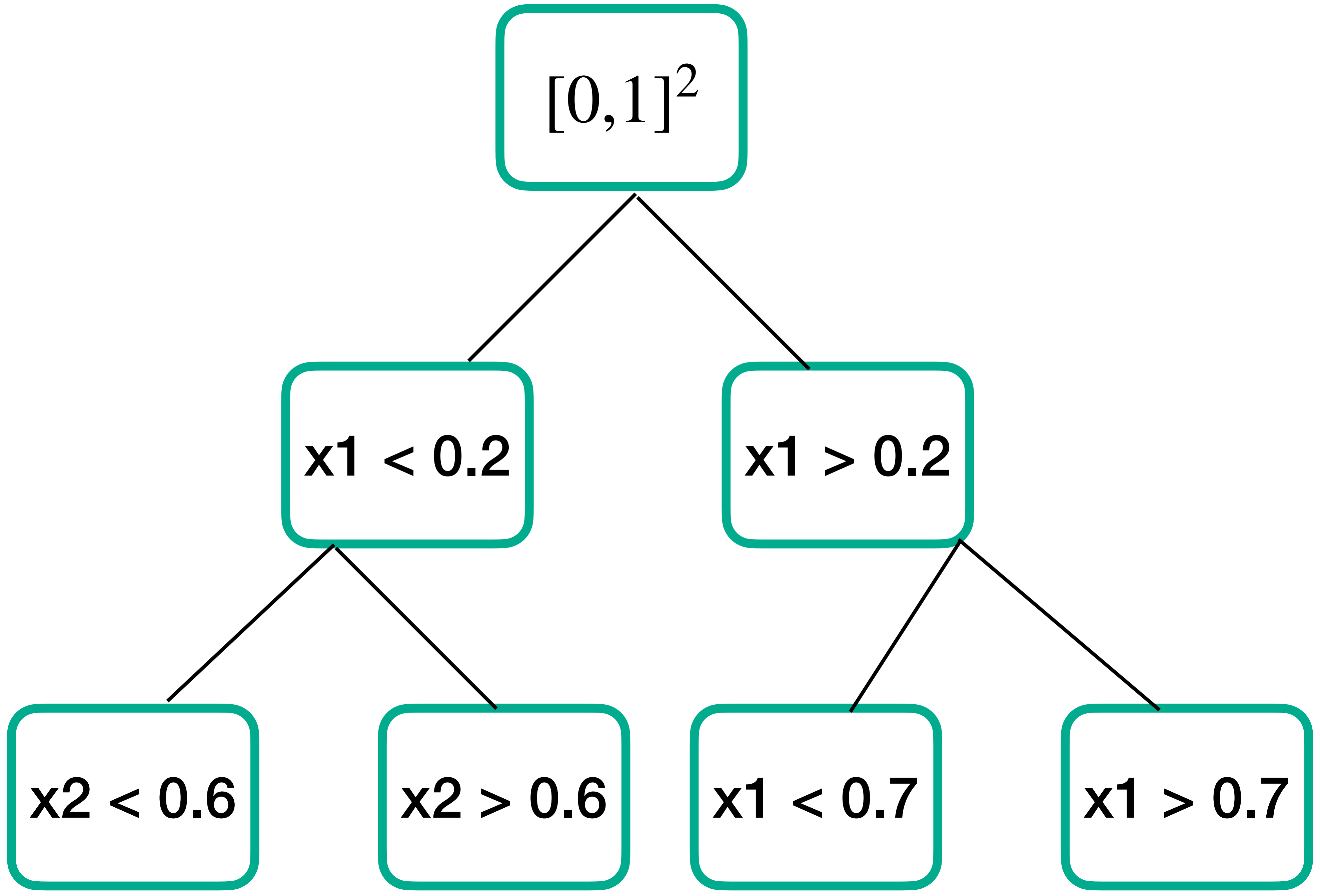


CART (Classification and regression tree) Split criterion: $\max_{Z,\beta} \frac{1}{n}\left(\|y - Z^{(0)}\hat{\beta}^{(0)}\|_2^2 - \|y - Z\beta\|_2^2 \right)$

Review of Regression Tress and Random Forests

Regression trees





CART (Classification and regression tree) Split criterion: $\max_{Z,\beta} \frac{1}{n}\left(\|y - Z^{(0)}\hat{\beta}^{(0)}\|_2^2 - \|y - Z\beta\|_2^2\right)$

CART-split criterion as OLS optimization

Nodes $C_1, \ldots, C_K$ with membership matrix $Z^{(0)}$ and node representatives (node means) $\hat{\beta}^{(0)} = (\beta_1^{(0)}, \ldots, \beta_K^{(0)})'$

To split the parent node C_k next, the CART-split criterion is equivalent to maximizing the following over $c, j,$ and β

$$\frac{1}{n}\left(\|Y - Z^{(0)}\hat{\beta}^{(0)}\|_2^2 - \|Y - Z(c,j)\beta\|_2^2 \right)$$

$Z(c, j)$ is the membership matrix for potential children nodes created by splitting C_k at variable j at cutoff c

CART-split criterion as OLS optimization

$$(\hat{c}, \hat{j}, \hat{\beta}) = \arg\max_{c,j,\beta} \frac{1}{n}\left(\|Y - Z^{(0)}\hat{\beta}^{(0)}\|_2^2 - \|Y - Z(c,j)\beta\|_2^2 \right)$$

New membership matrix: $Z = Z(\hat{c}, \hat{j})$

New node representatives: $\hat{\beta} = (Z'Z)^{-1}Z'Y$

DART-split criterion using GLS loss

Replace CART split criterion, a global OLS loss

$$(\hat{c}, \hat{j}, \hat{\beta}) = \arg\max_{c,j,\beta} \frac{1}{n} \left(\|Y - Z^{(0)}\hat{\beta}^{(0)}\|_2^2 - \|Y - Z(c,j)\beta\|_2^2 \right)$$
$$= \arg\min_{c,j,\beta} \frac{1}{n} \|Y - Z(c,j)\beta\|_2^2$$

with *Dependency-adjusted Regression Tree (DART)-split criterion* a global GLS loss

$$(\hat{c}, \hat{j}, \hat{\beta}) = \arg\min_{c,j,\beta} \frac{1}{n} (y - Z(c,j)\beta)' \Sigma^{-1} (y - Z(c,j)\beta)$$

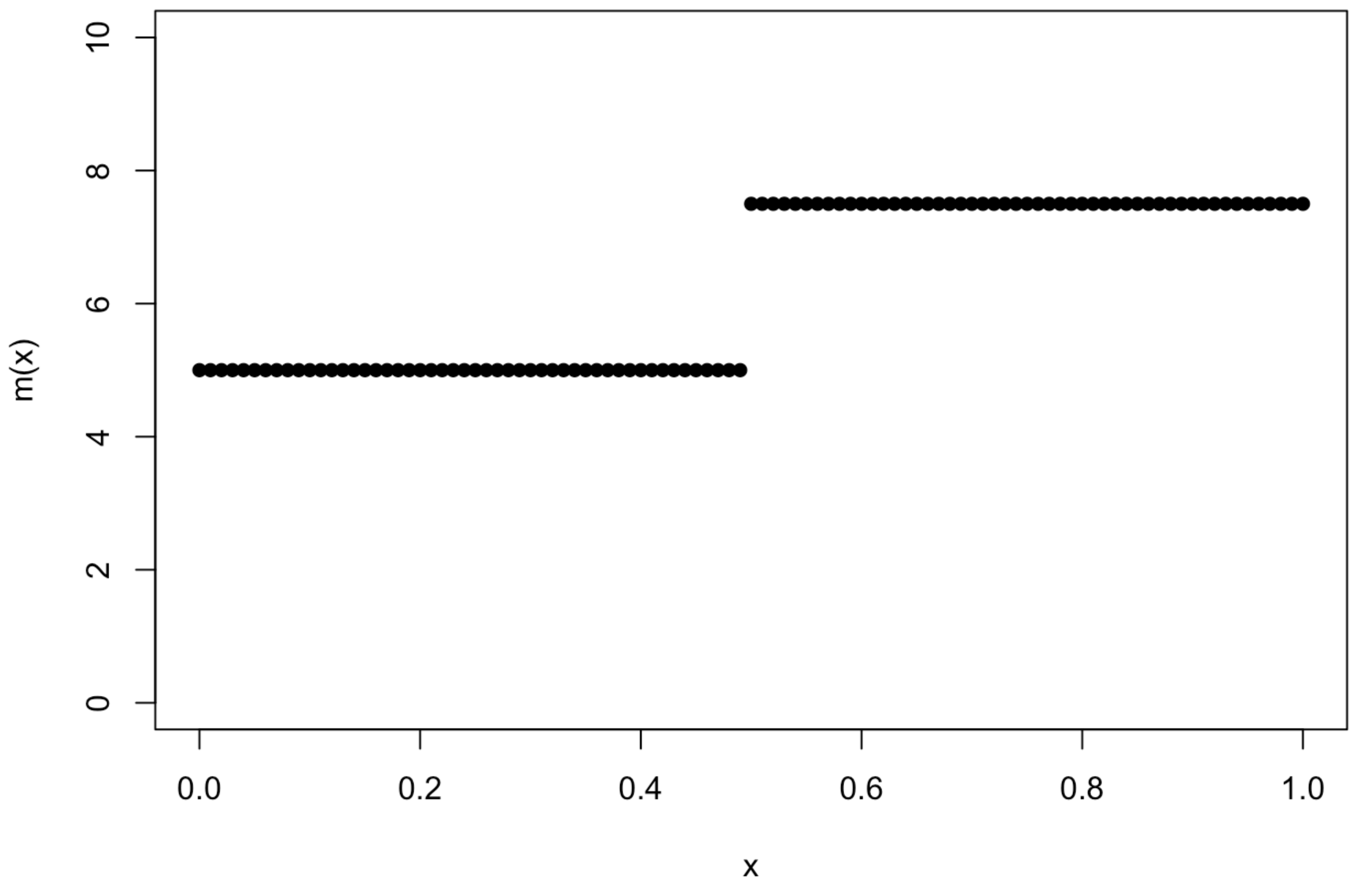
DART-split criterion using GLS loss

Dependency-adjusted Regression Tree (DART)-split criterion, a global GLS loss

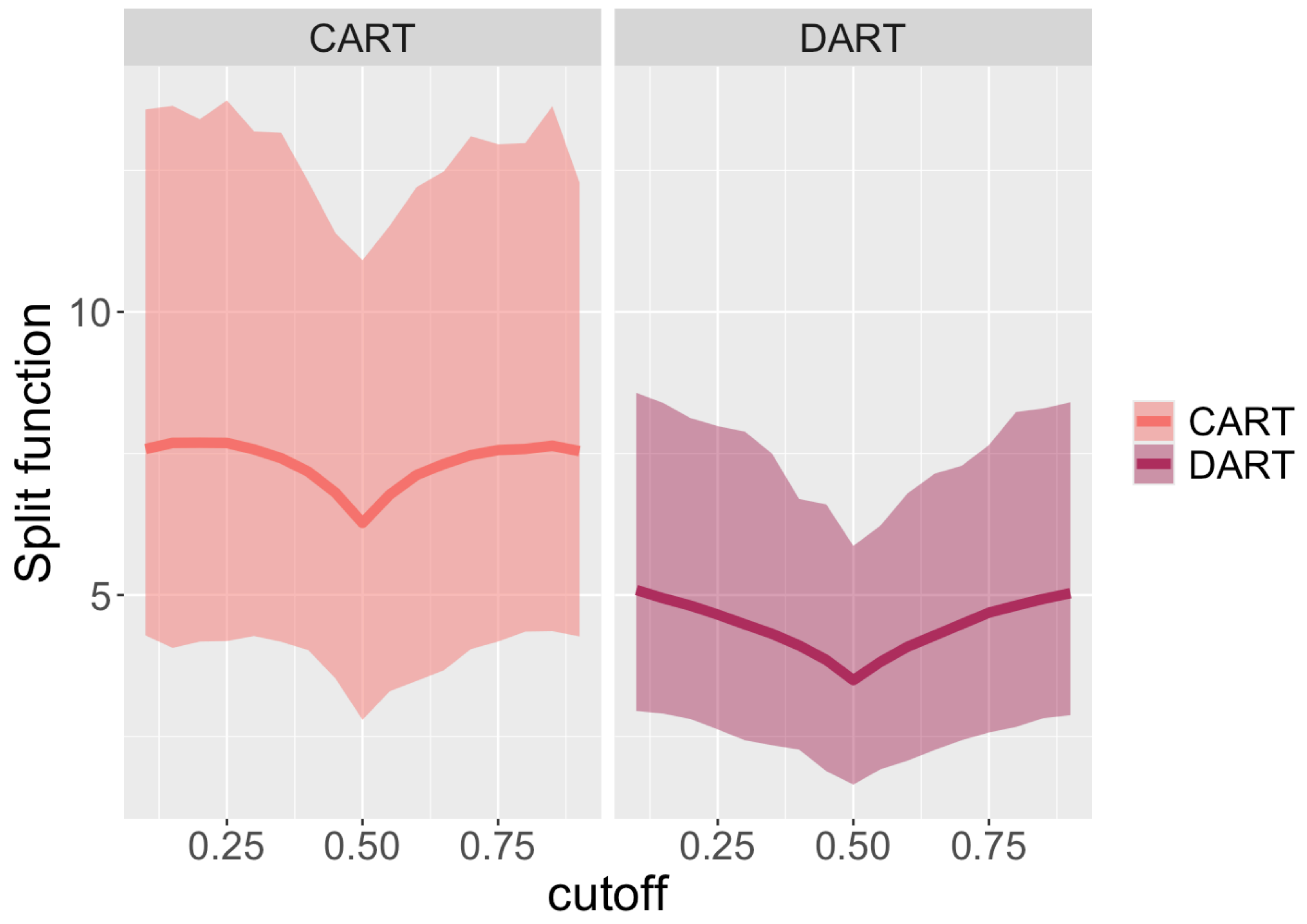
$$(\hat{c}, \hat{j}, \hat{\beta}) = \arg\min_{c,j,\beta} \frac{1}{n}(y - Z(c,j)\beta)'\Sigma^{-1}(y - Z(c,j)\beta)$$

In practice, Σ is replaced by an estimate $\widehat{\Sigma}$, the working covariance matrix

GLS vs OLS tree for dependent data

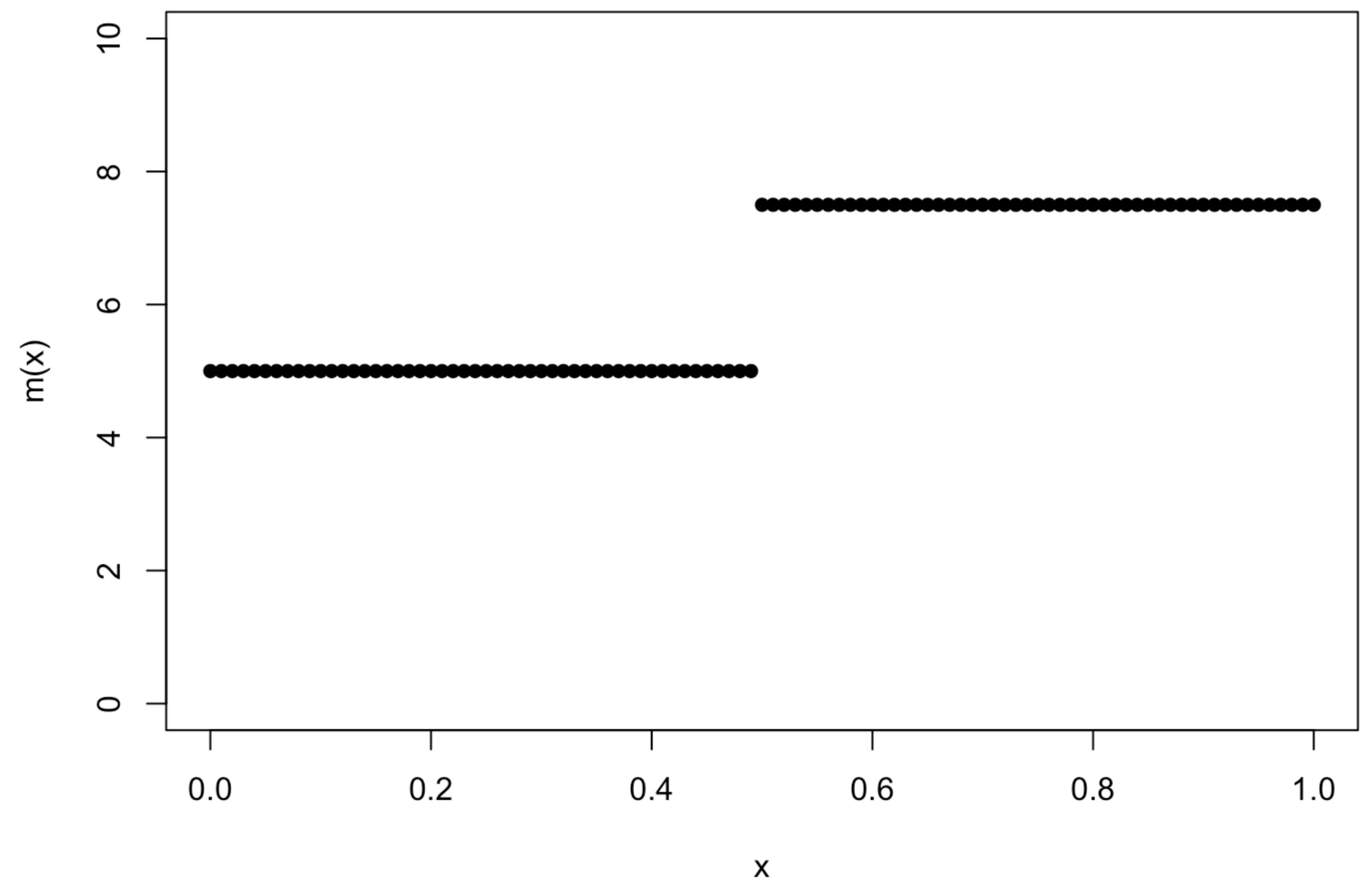


True $m(x)$: Discontinuity at 0.5

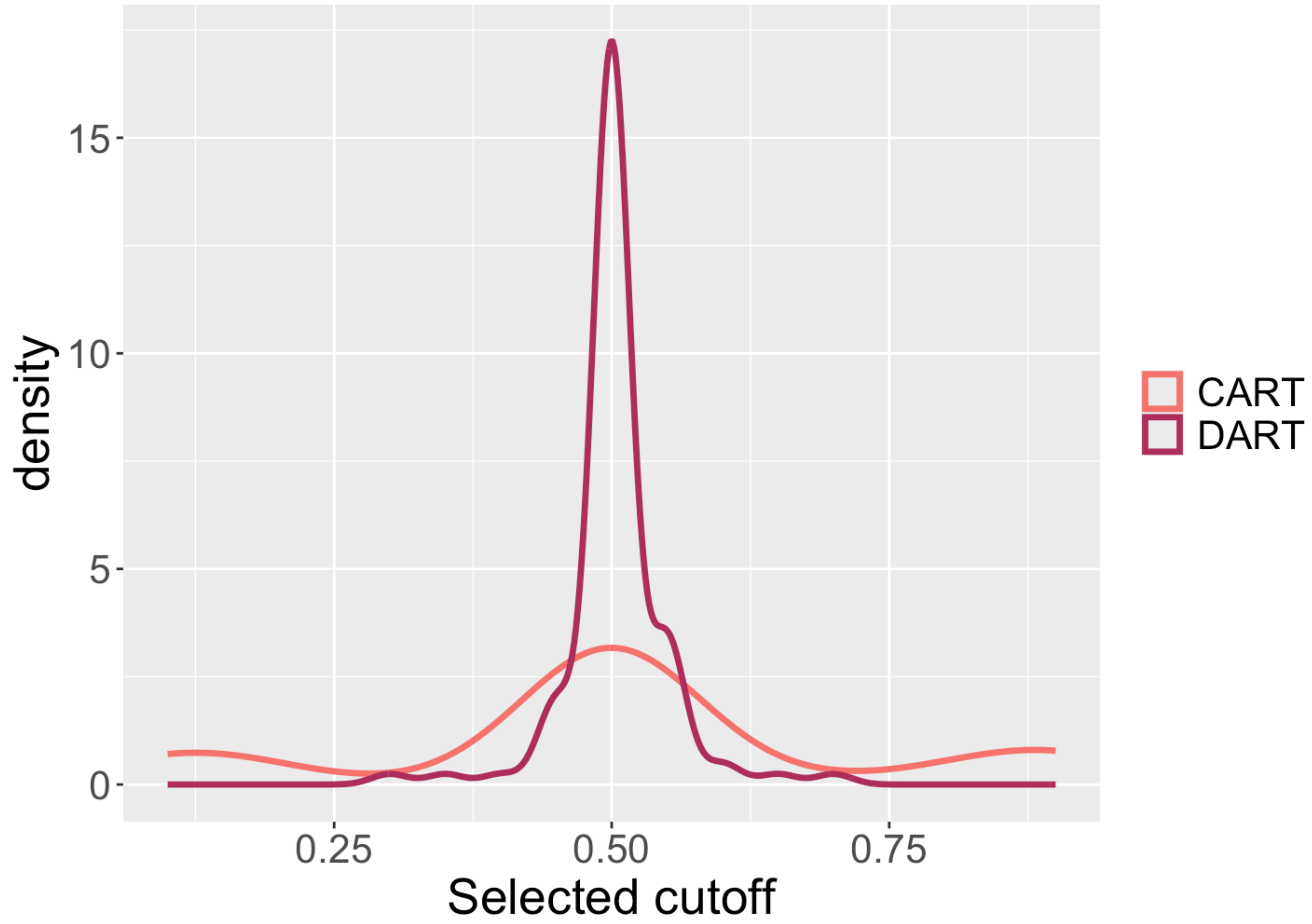


CART and DART loss as function of cutoff for 100 datasets

GLS vs OLS tree for dependent data



True $m(x)$: Discontinuity at 0.5



Density of selected cutoffs minimizing CART and DART loss over 100 datasets

GLS-style regression tree

Build the tree by sequentially splitting nodes

- Maximize the DART split criterion for splitting a node
- Update the membership matrix to reflect the current set of nodes
- Repeat till a stopping criterion is met (e.g., minimum nodesize)

Membership matrix $\hat{Z}$ corresponding to final set of nodes

Final set of node representatives $\hat{\beta}_{GLS} = (\hat{Z}'Q\hat{Z})^{-1}\hat{Z}'QY$

$Q = \widehat{\Sigma}^{-1}$ is the working precision matrix

Both splitting of nodes and representative assignment uses correlation among all data points

Trees to forest

RF estimate is an average over several tree estimates

Each tree in RF uses a resample of the data $P_t Y$ where P_t is the resampling matrix

- Under dependence, this will end up resampling correlated data
- Leads to singularity of the GP covariance matrix

Correlation adjusted resampling

Regression tree with a resample $P_t Y$ uses the OLS loss function $||P_t Y - P_t Z\beta||^2$

GLS loss with Y and Z using a working precision matrix Q is equivalent to OLS loss with $\tilde{Y} = Q^{1/2}Y$ and $\tilde{Z} = Q^{1/2}Z$.

Immediate extension for resampling: Use the tree-specific DART split-criterion

$$||P_t\tilde{Y} - P_t\tilde{Z}\beta||^2$$

Only needs the Cholesky factor $Q^{1/2} = \widehat{\Sigma}^{-1/2}$

We essentially resample the contrasts (prewhitened data) $\tilde{Y} = Q^{1/2}Y$

RF-GLS estimation summary

Create n_{tree} many resampling matrices $P_1, \ldots, P_{n_{tree}}$

For the t^{th} resampling matrix P_t, build GLS-style tree using DART split criterion

Final set of nodes and node representatives $\hat{\beta}^{(t)}_{GLS}$

Tree-estimate of $m(x)$ is the k^{th} component of $\hat{\beta}^{(t)}_{GLS}$ if $x \in k^{th}$ node of the t^{th} tree

RF-GLS estimate of $m(x)$ is the average of all tree-specific estimates

Predictions with RF-GLS

Recall: When $m(x) = x'\beta$, predictive distribution at a new location s_0 is given by

$$Y(s_0) \mid Y, \theta, \beta = N\left(\mu(s_0), \sigma^2(s_0)\right)$$

Conditional (kriging) mean: $\mu(s_0) = X'(s_0)\hat{\beta} + C(s_0, S)\Sigma^{-1}(Y - X\hat{\beta})$

Conditional (kriging) variance: $\sigma^2(s_0) = C(s_0, s_0) + \tau^2 - C(s_0, S)\Sigma^{-1}C(S, s_0)$

Predictions with RF-GLS

For RF-GLS, predictive distribution at a new location s_0 is given by

$$Y(s_0) \mid Y, \theta, \beta = N\left(\mu(s_0), \sigma^2(s_0)\right)$$

Conditional (kriging) mean: $\mu(s_0) = \widehat{m}(X(s_0)) + C(s_0, S)\Sigma^{-1}(Y - \widehat{m}(X))$

Conditional (kriging) variance: $\sigma^2(s_0) = C(s_0, s_0) + \tau^2 - C(s_0, S)\Sigma^{-1}C(S, s_0)$

Immediate extension for RF-GLS

Advantage of RF-GLS being embedded in the spatial mixed model based framework

Practical implementation

Spatial parameter estimation:

Estimate the spatial parameters in Σ using the residuals $y_i - \widehat{m_{init}}(X_i)$ using RF to get initial estimate of m

Speedup using NNGP:

Only one time evaluation of the Cholesky factor $\Sigma^{-1/2}$

- Requires $O(n^3)$ computation
- We use $Q = \tilde{\Sigma}^{-1}$ where $\tilde{\Sigma}$ is the Nearest Neighbor Gaussian Process (NNGP) covariance matrix
- NNGP requires $O(n)$ time and directly gives $Q^{1/2} = \tilde{\Sigma}^{-1/2}$

RandomForestsGLS R-package

Model estimation using the *RFGLS_estimate_spatial* function

Mean function prediction using the *RFGLS_predict* function

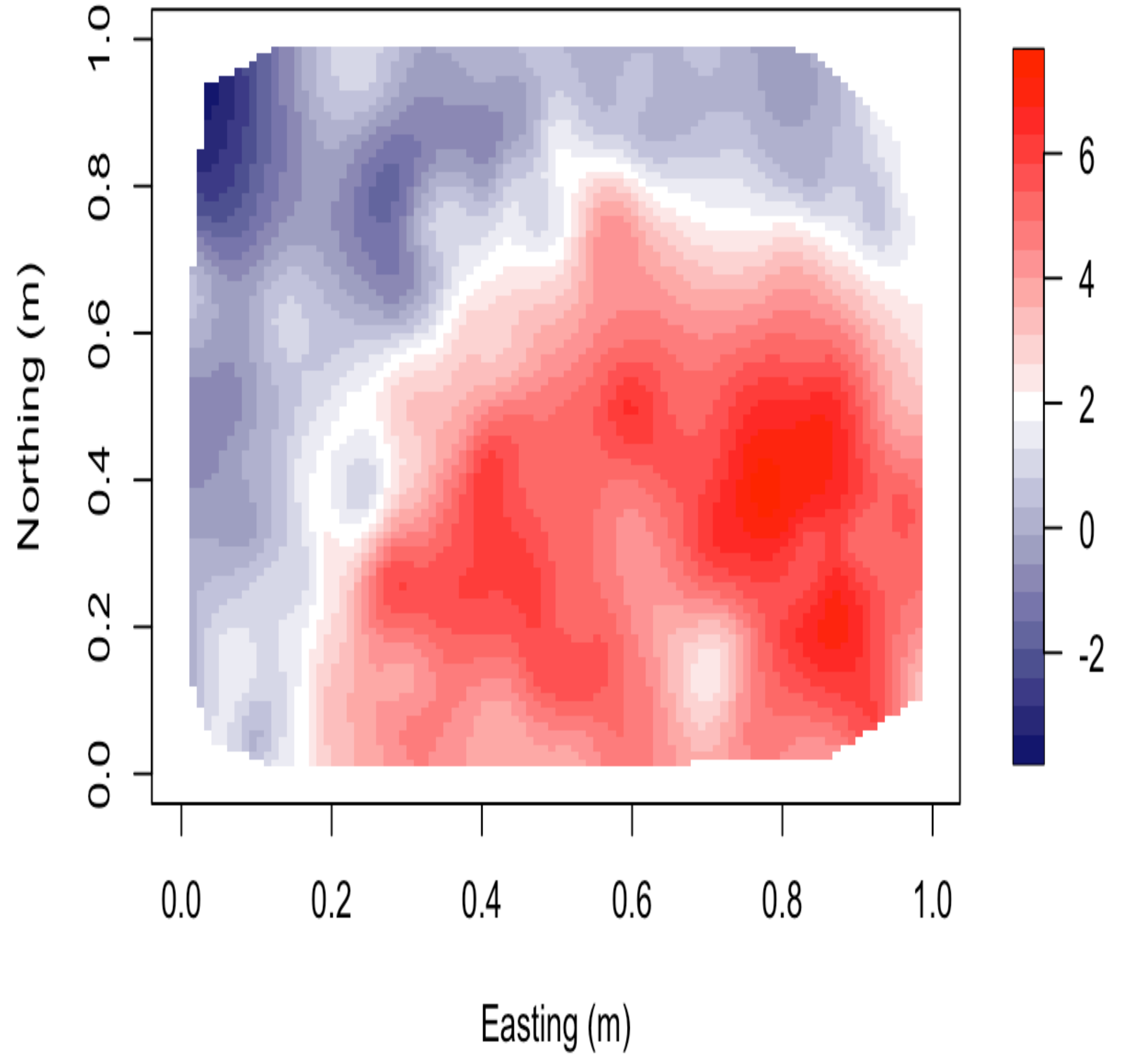
Spatial predictions of the response using the *RFGLS_predict_spatial* function

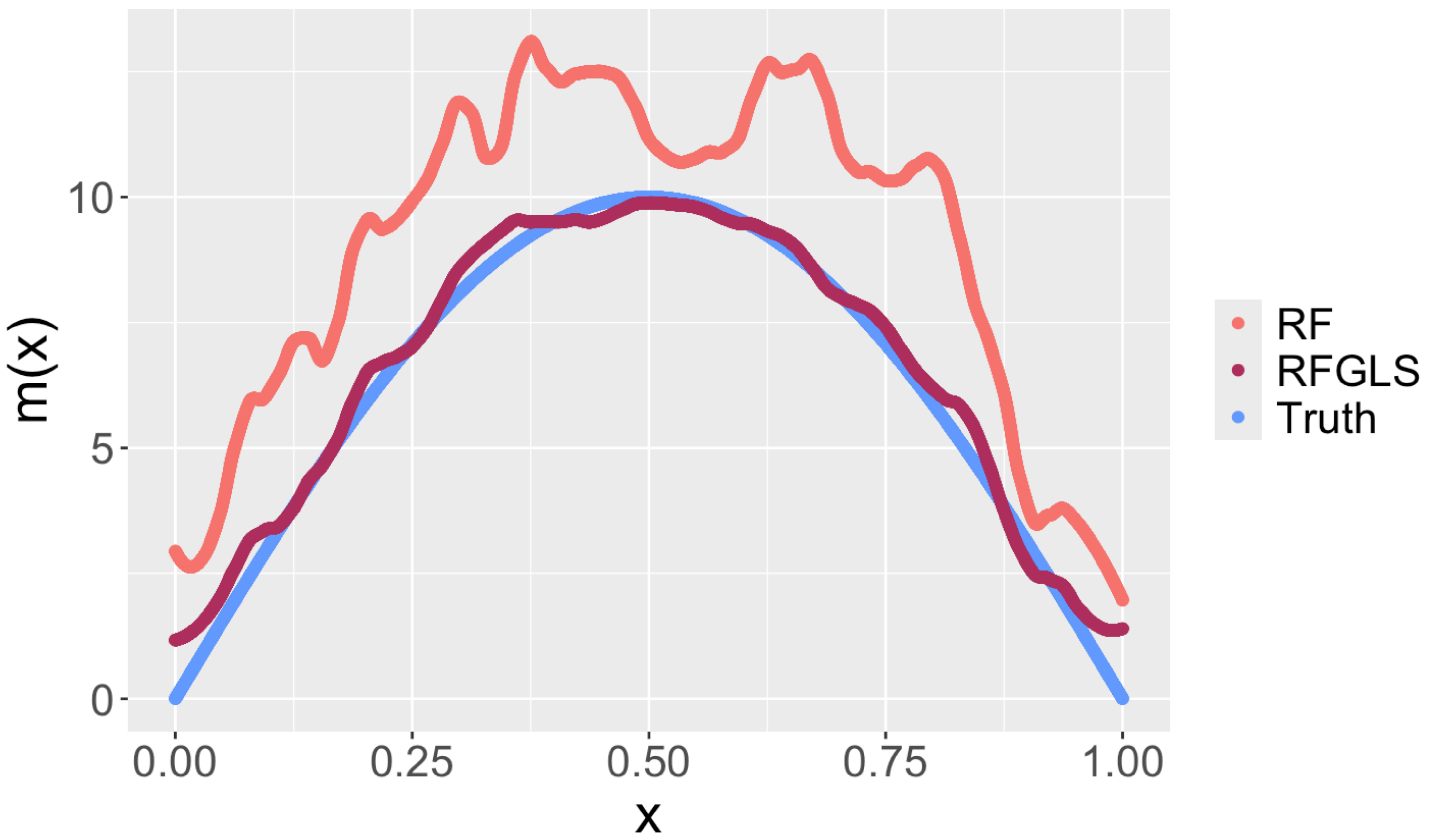
Available on CRAN: https://cran.r-project.org/web/packages/RandomForestsGLS/

Vignette: *How to use RandomForestsGLS*

RF vs RF-GLS for spatially dependent data

$m(x) = 10sin(\pi x)$





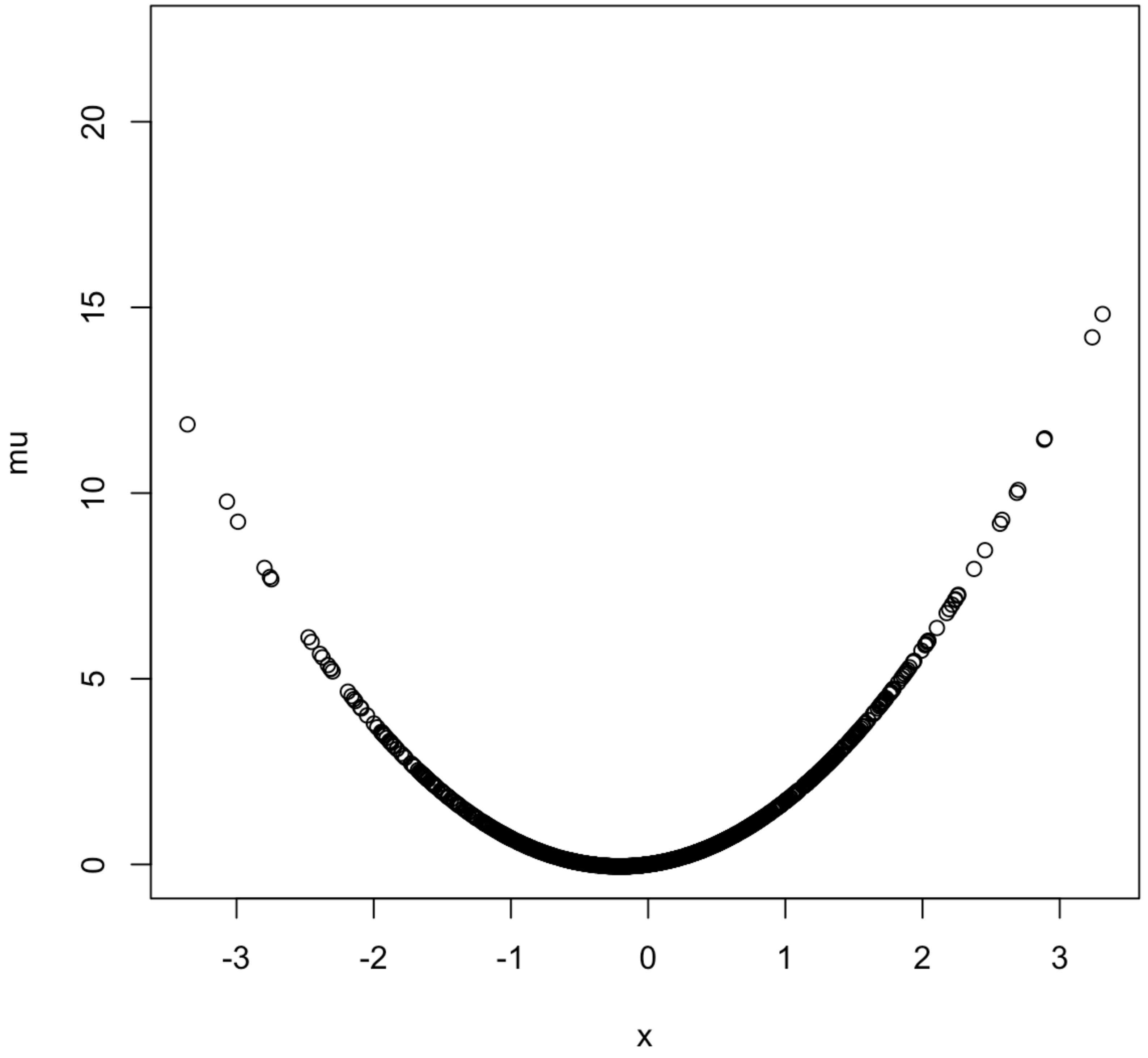
spatially correlated errors

$\widehat{m}(x)$ from RF and RF-GLS

Computational strategies

RFGLS_estimate_spatial can be slow for moderate-sized datasets

Example: $n = 1000$

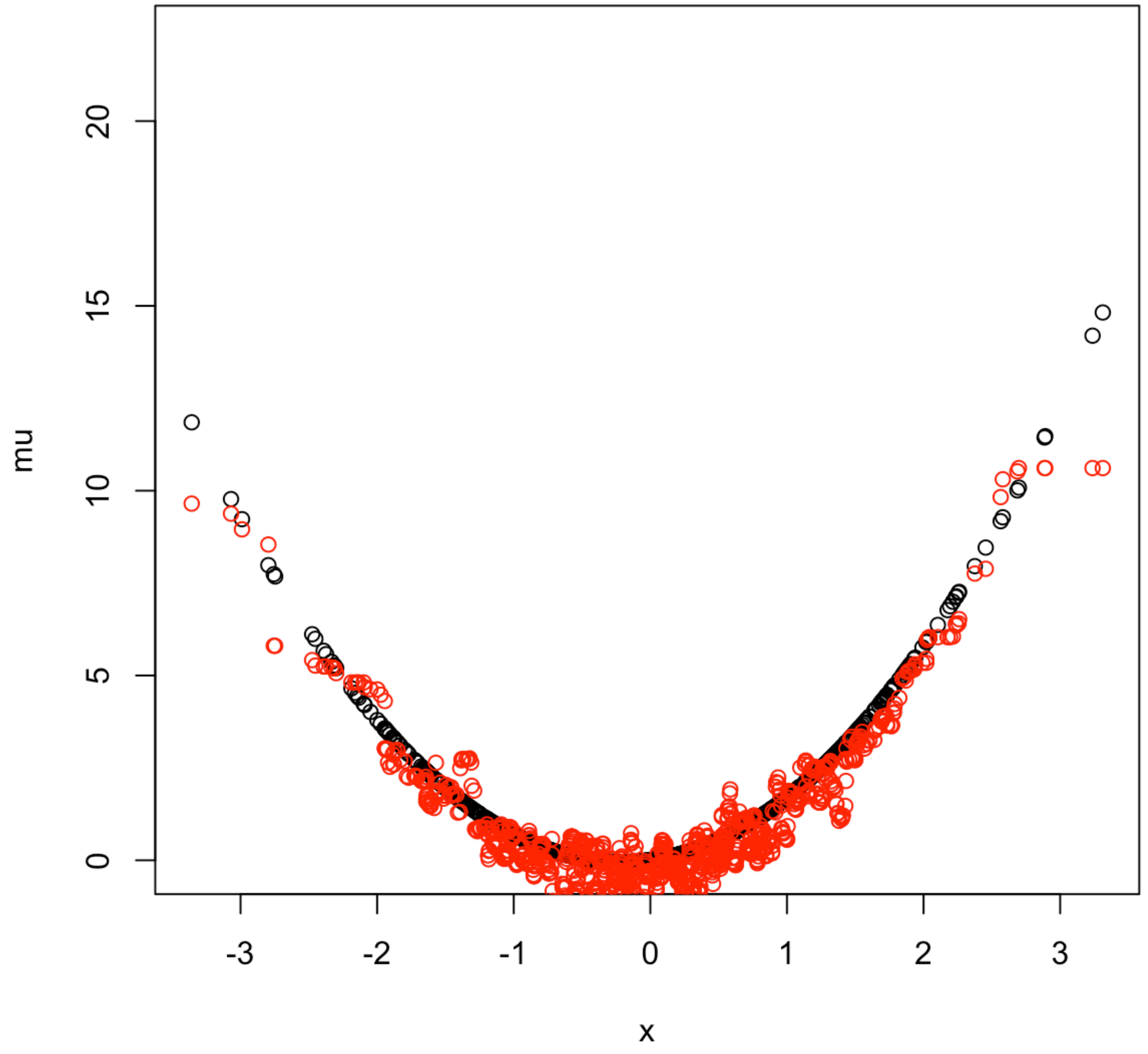


True mean function

Computational strategies

RFGLS_estimate_spatial can be slow for moderate-sized datasets

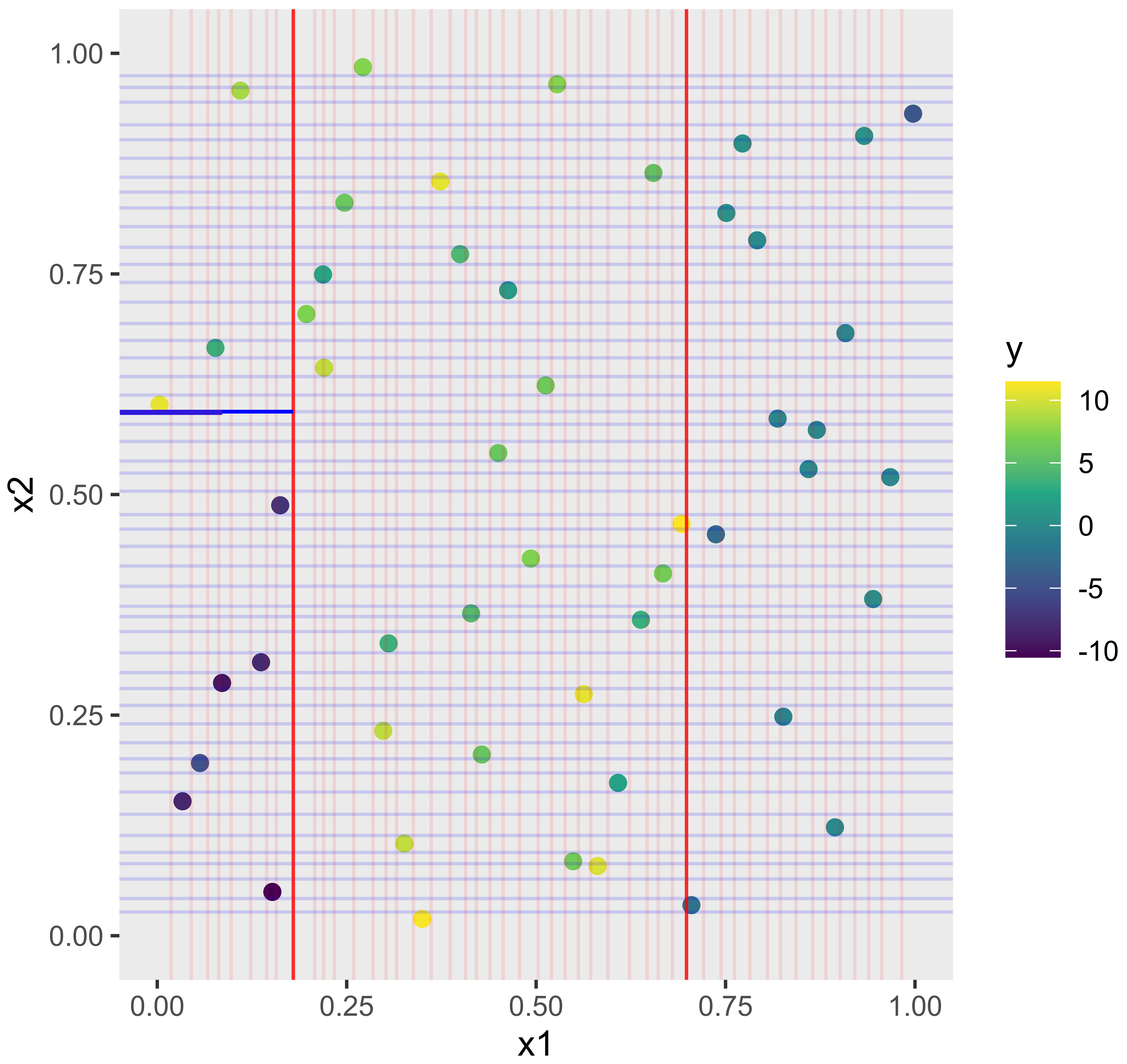
Example: $n = 1000$

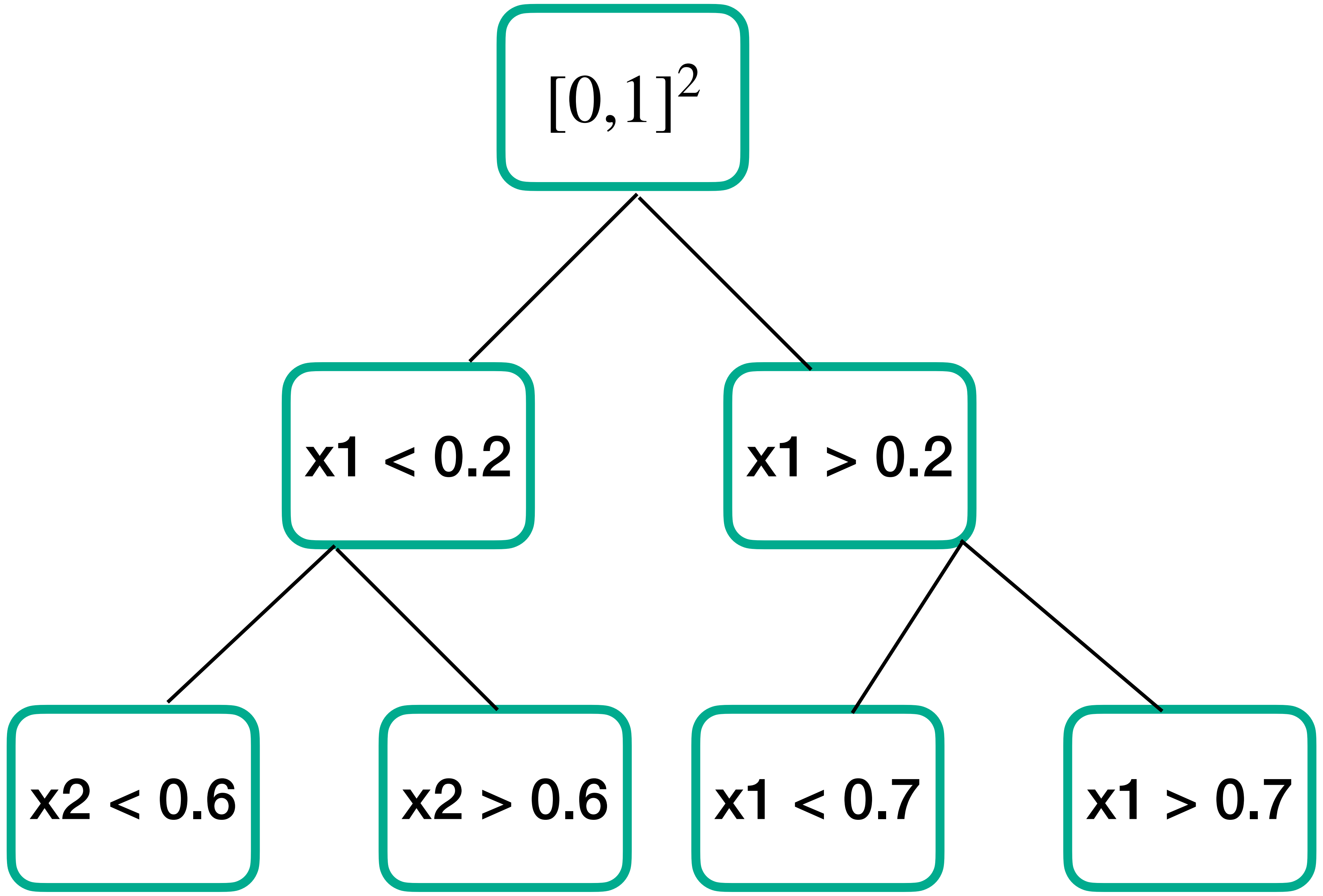


Fit (red) from RF-GLS, Running time: 35 mins

Computational strategies: Rounding

Trees in random forests are built by searching through gaps in covariates

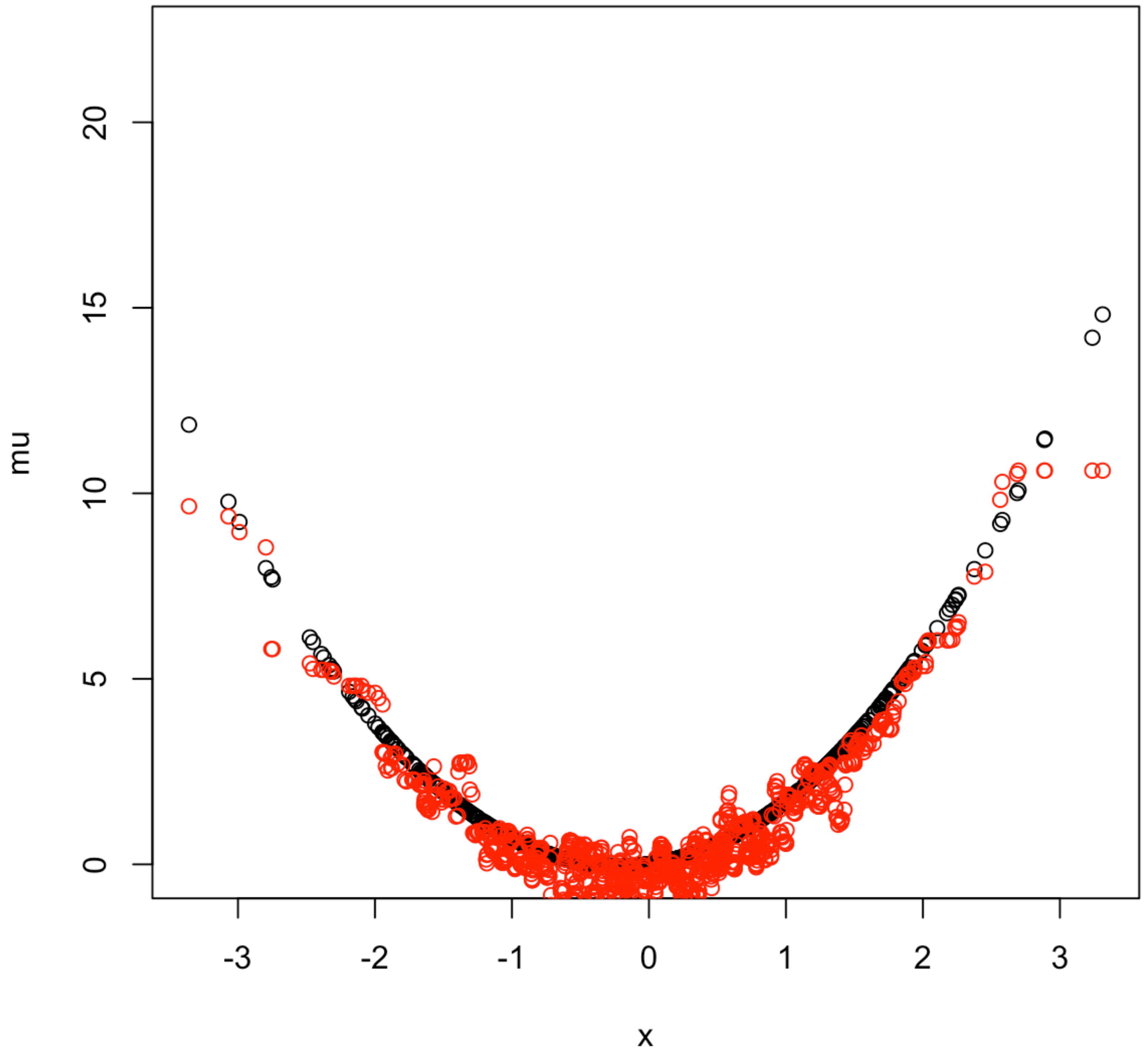




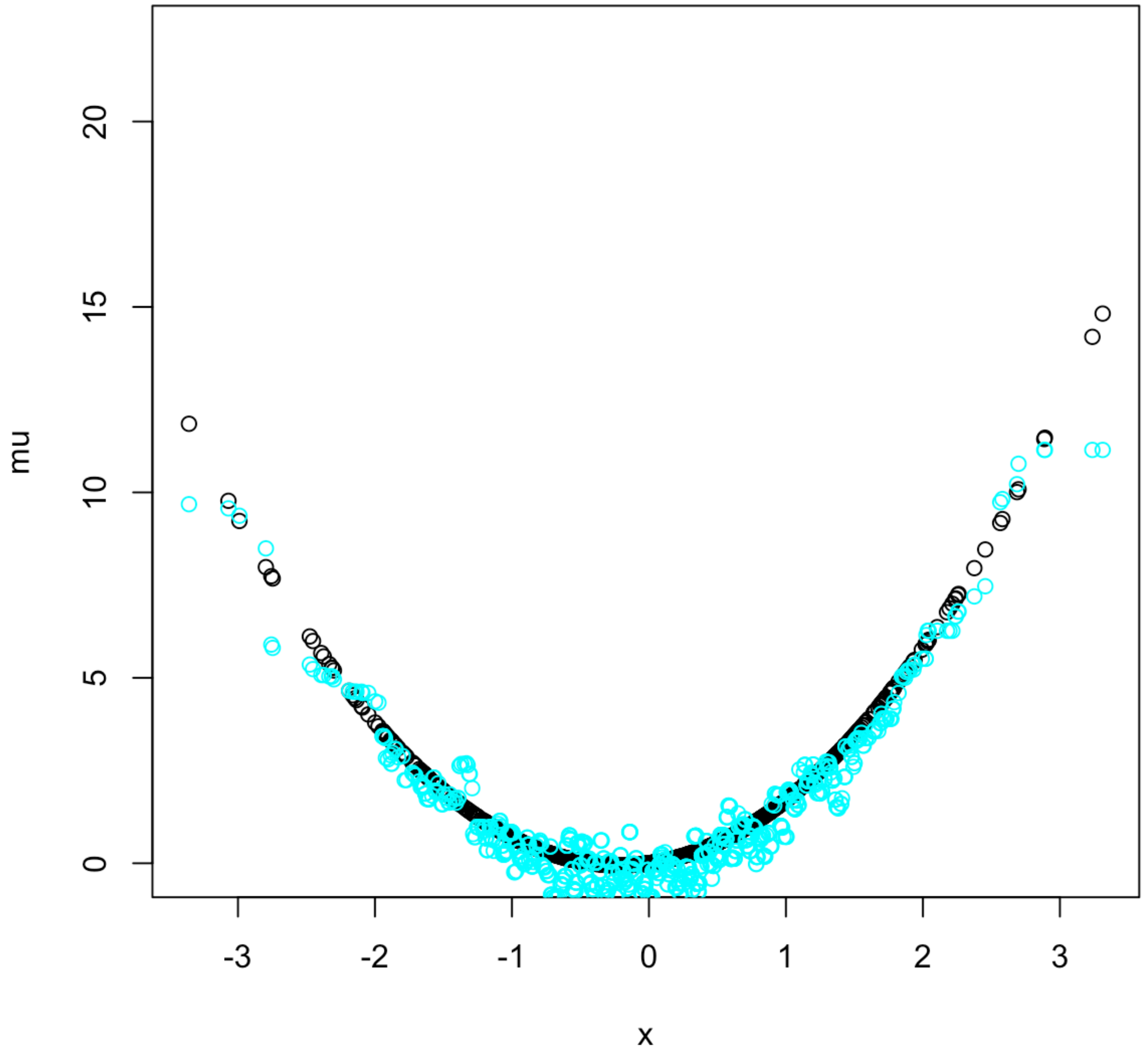
Computational strategies: Rounding

Trees in random forests are built by searching through gaps in covariates

Rounding the covariates reduce the number of gaps and running time



Using original x, 999 gaps, Running time: 35 mins

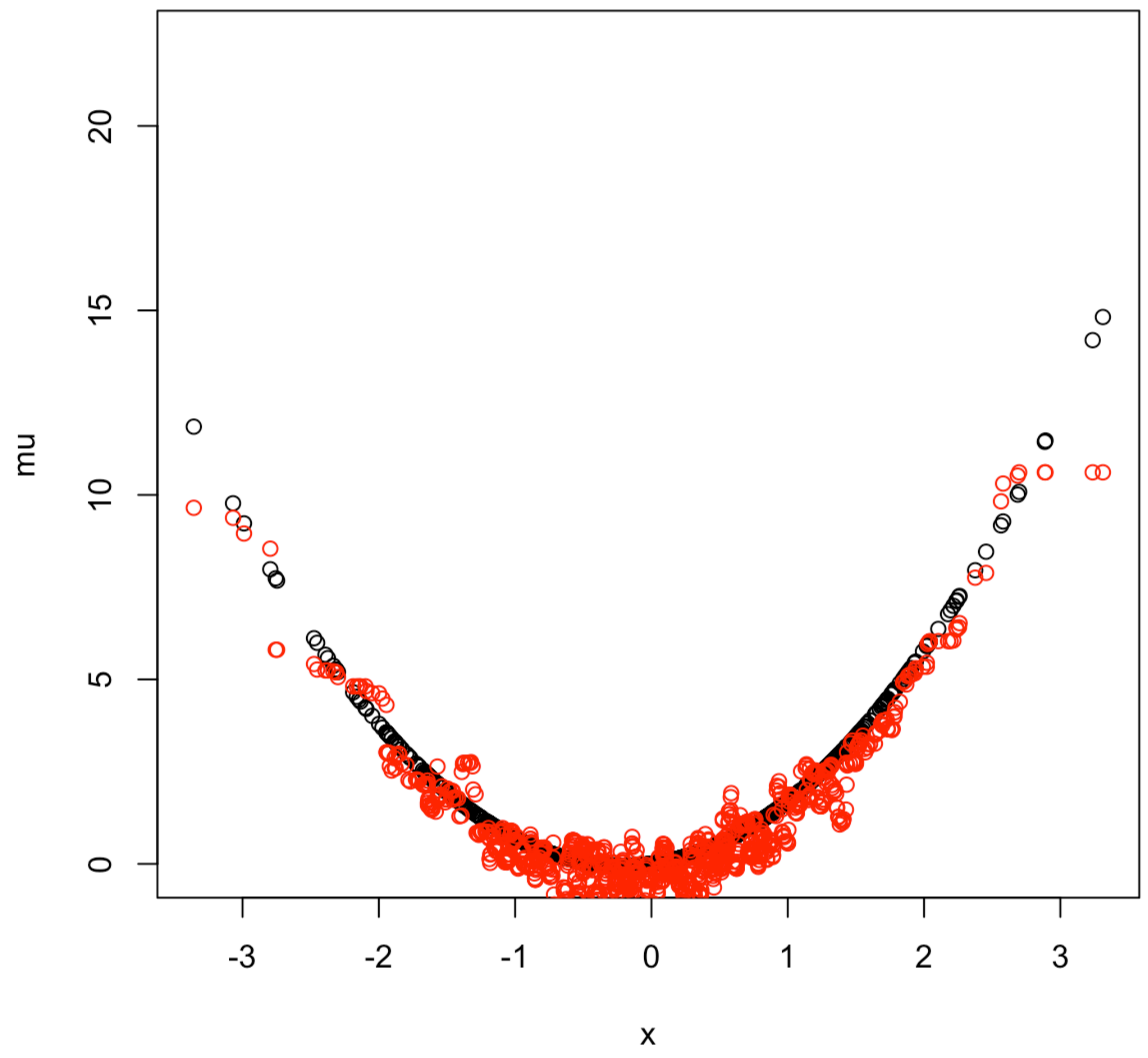


Rounded x to 2 decimal places, 374 gaps, Running time: 8 mins

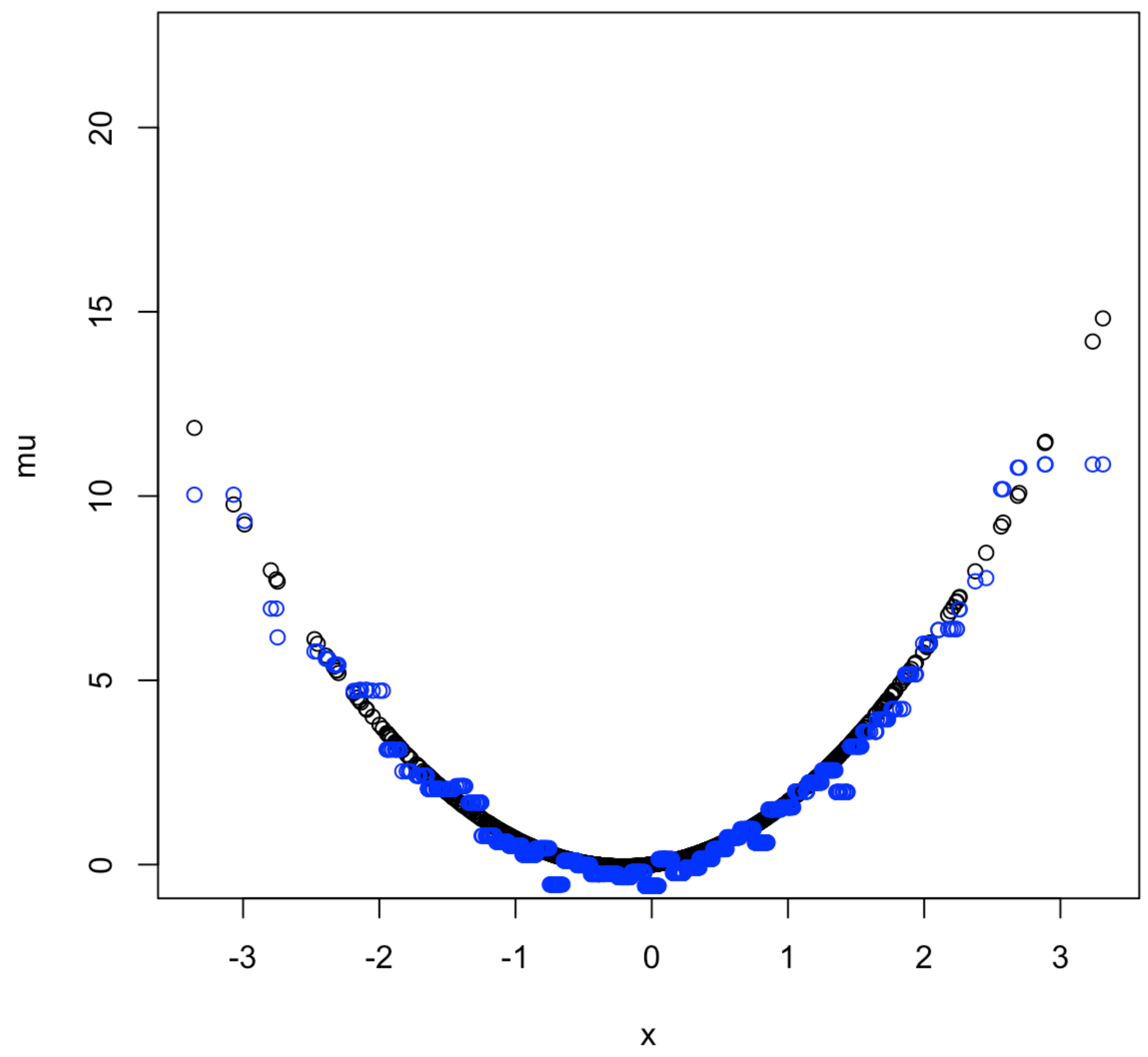
Computational strategies: Rounding

Trees in random forests are built by searching through gaps in covariates

Rounding the covariates reduce the number of gaps and running time



Using original x, 999 gaps, Running time: 35 mins



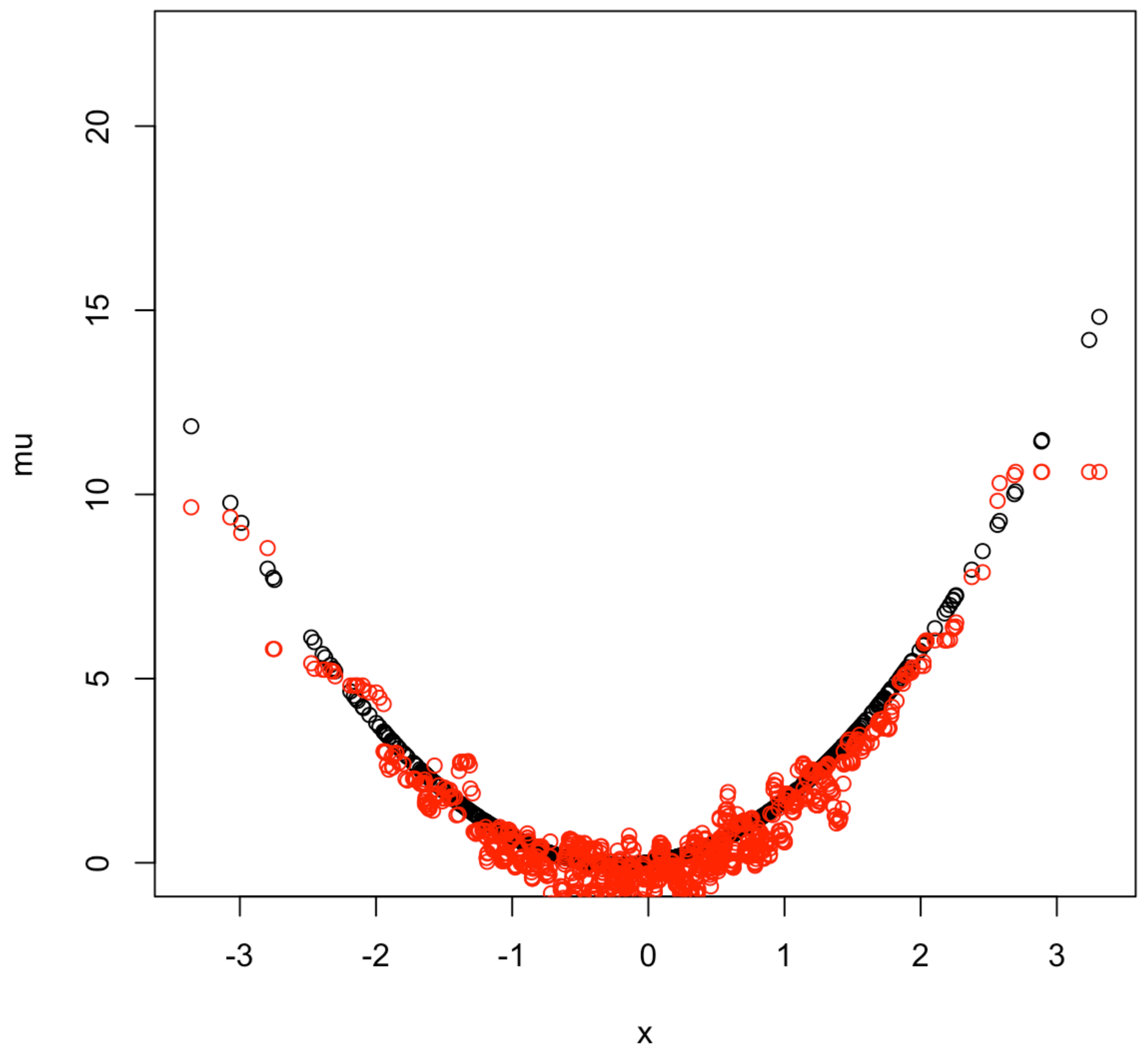
Rounded x to 1 decimal places, 60 gaps, Running time: 2 mins

Computational strategies: Binning

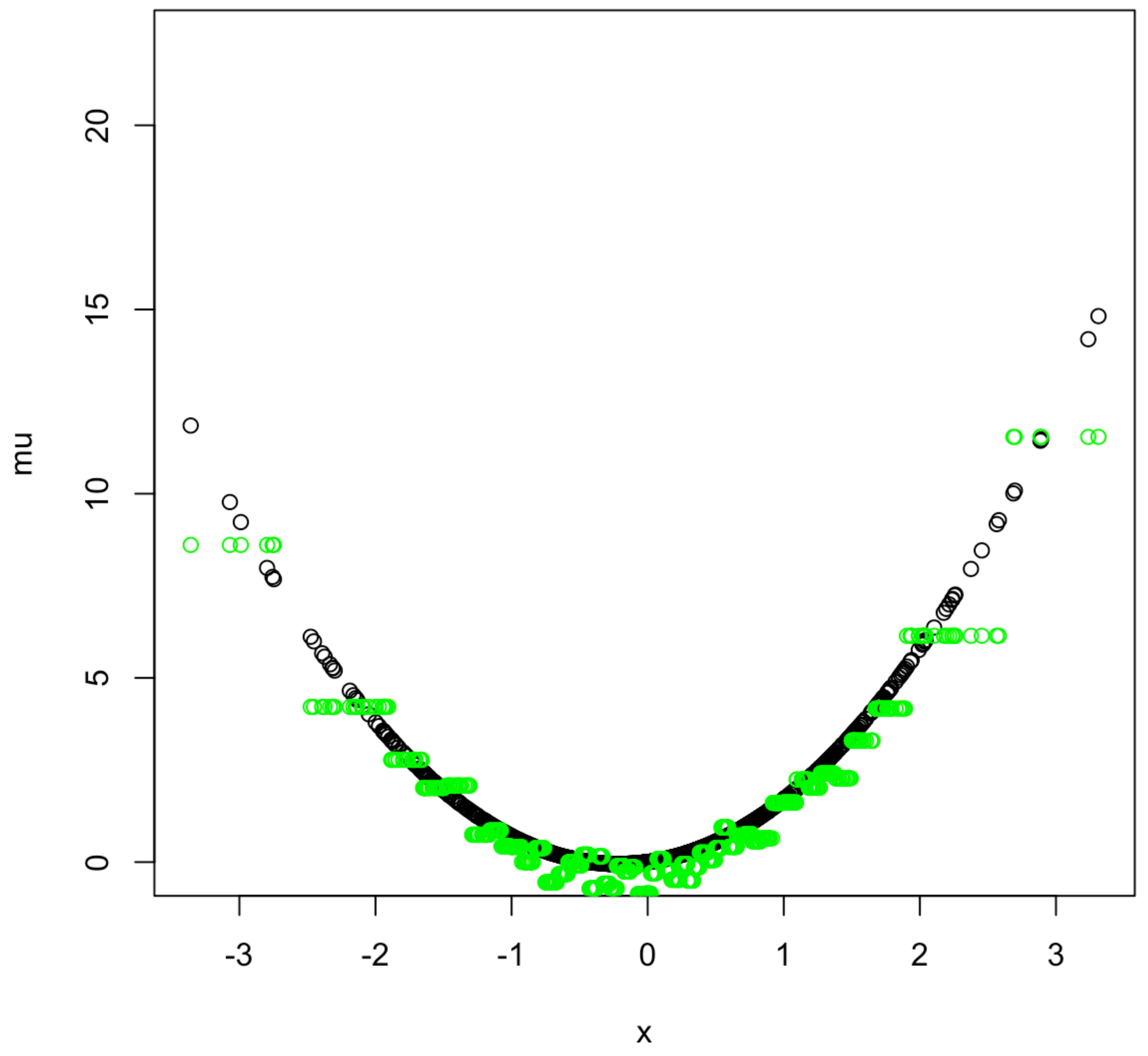
Rounding to fixed number of decimal digits is binning at fixed width

Alternatively, one can bin to quantiles of X (e.g., bin to nearest even quantile)

Bins are of variable widths determined by the distribution of X



Using original x, 999 gaps, Running time: 35 mins

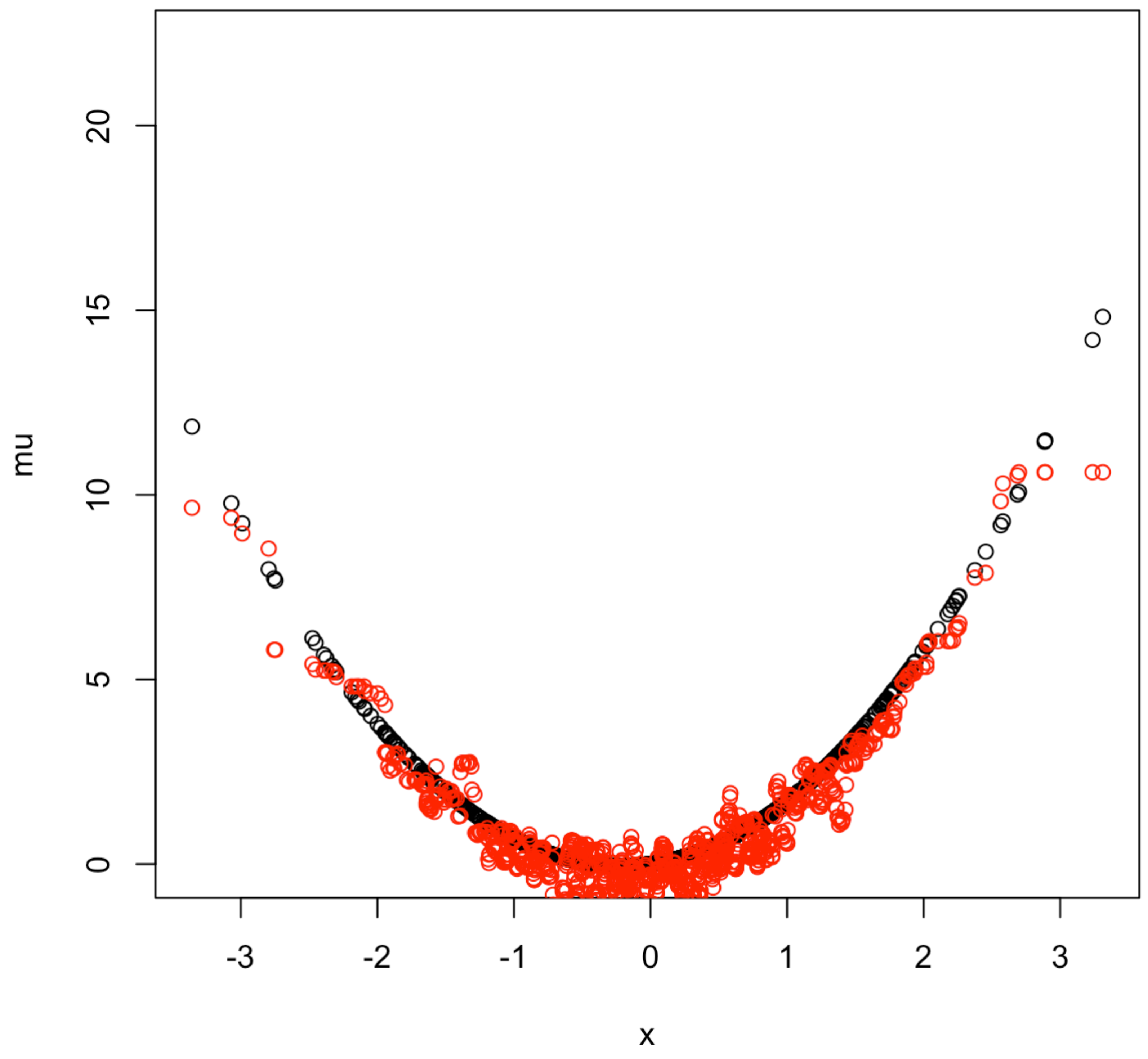


Binned to nearest even quantile, 50 gaps, Running time: 2 mins

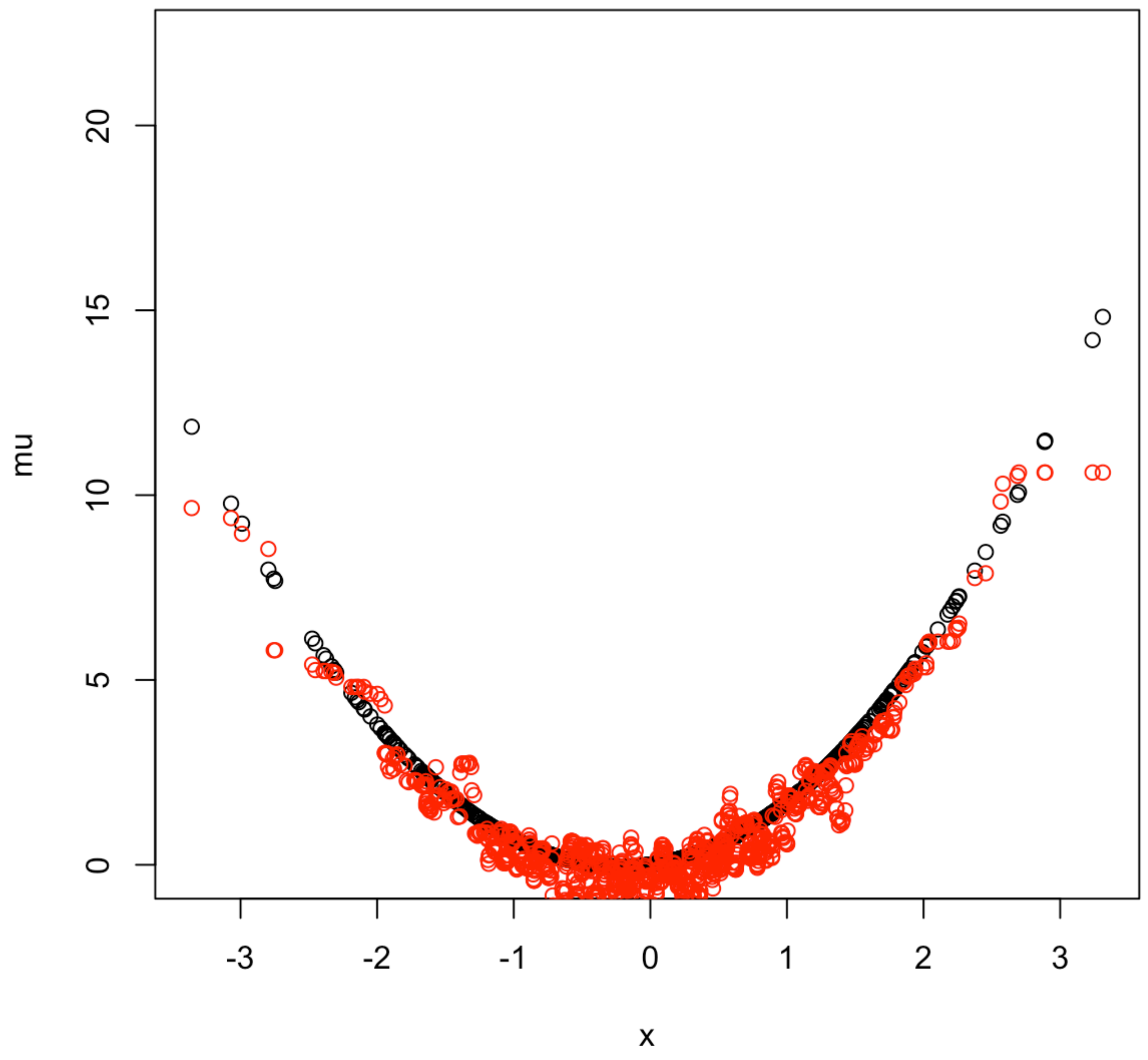
Computational strategies: Parallelization

RFGLS_estimate_spatial allows parallel computations

Number of cores can be set by the h argument (default is $h = 1$)



$h = 1$ (No parallelization), Running time: 35 mins



$h = 10$, Running time: 23 mins

Computational strategies: Parallelization

RFGLS_estimate_spatial allows parallel computations

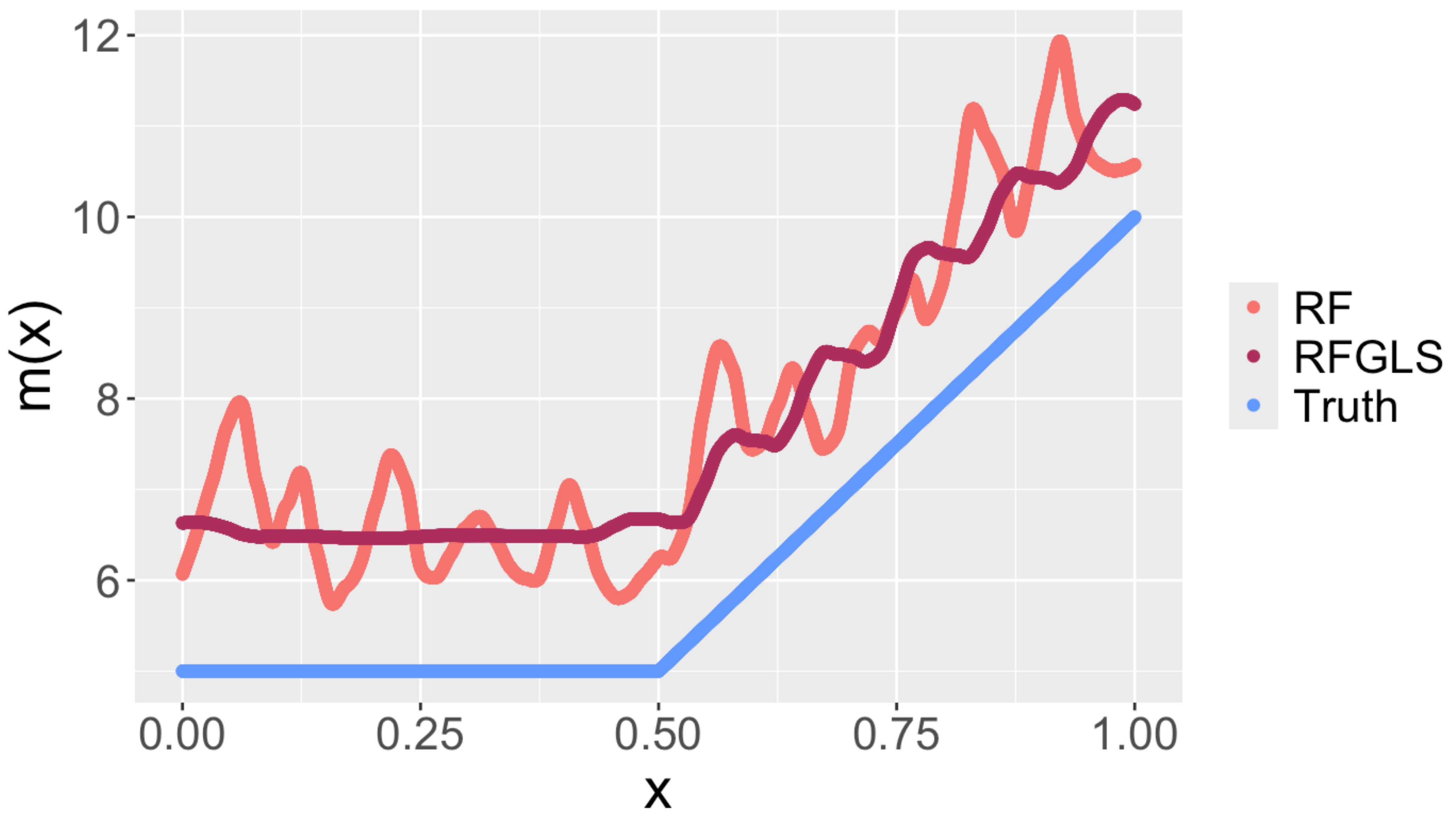
- Number of cores can be set by the h argument (default is $h = 1$)
- h needs to be strictly less than the total number of cores
- Only recommended for larger datasets

Prediction is very fast and do not require parallelization

Mean shift

Mean function estimates can sometimes have a constant shift

- Occurs when the locations are densely packed
- Insight: Even in linear regression, GLS estimators may not identify the intercept under in-fill sampling

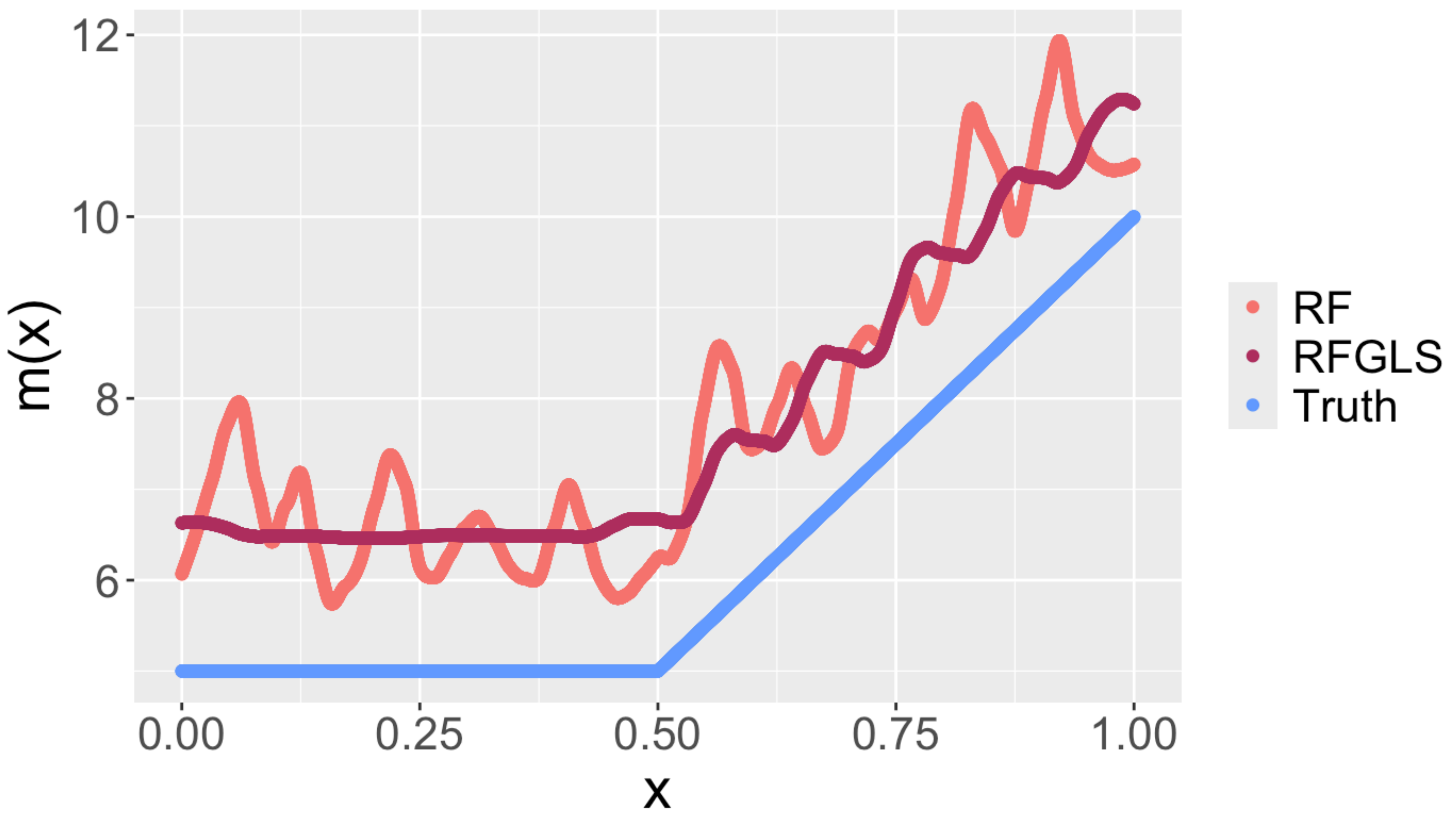


True function and estimates
for data at 500 locations in $[0,1] \times [0,1]$

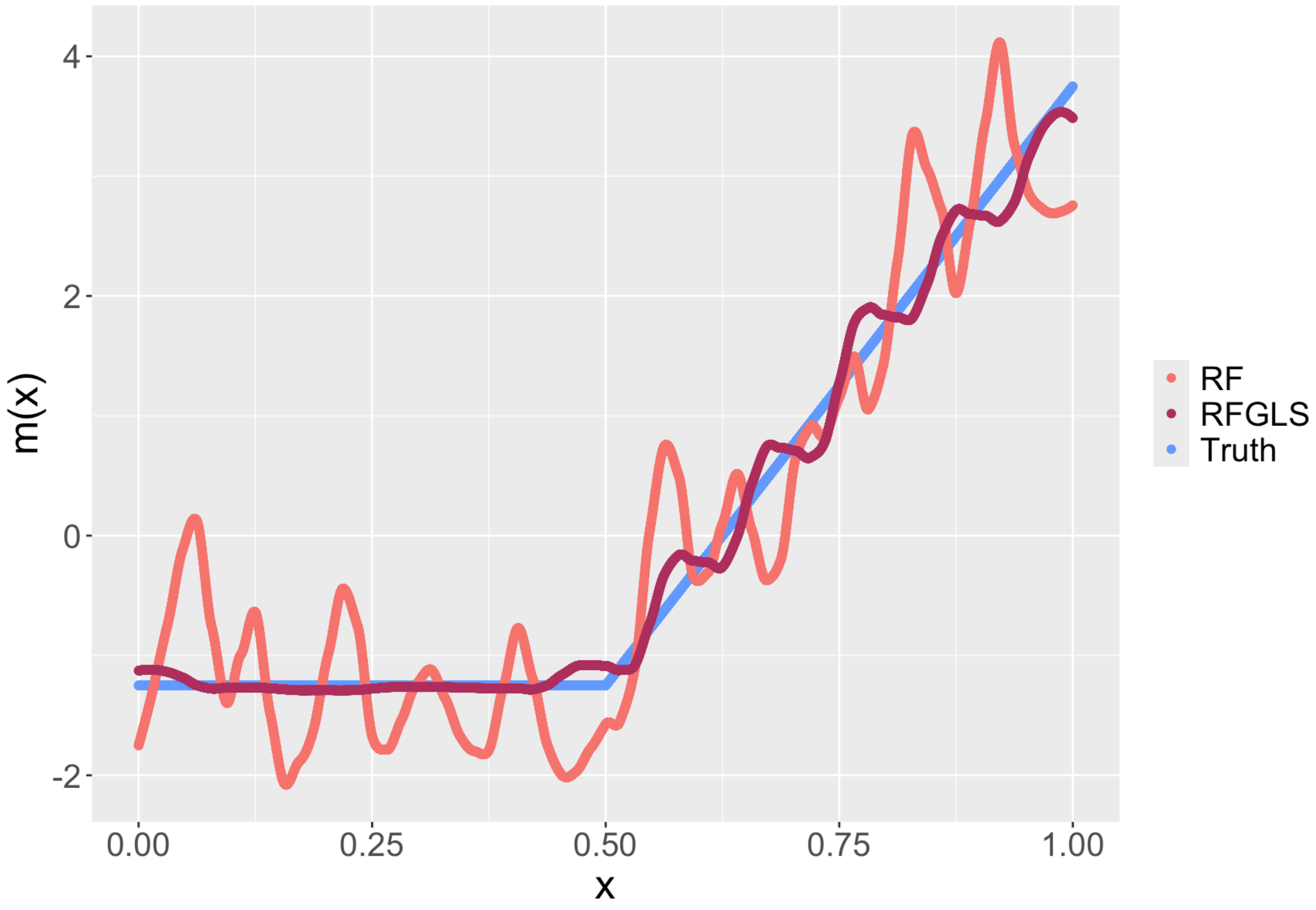
Mean shift

Mean function estimates can sometimes have a constant shift

One can thus look at centered estimates



True function and estimates
for data at 500 locations in $[0,1] \times [0,1]$

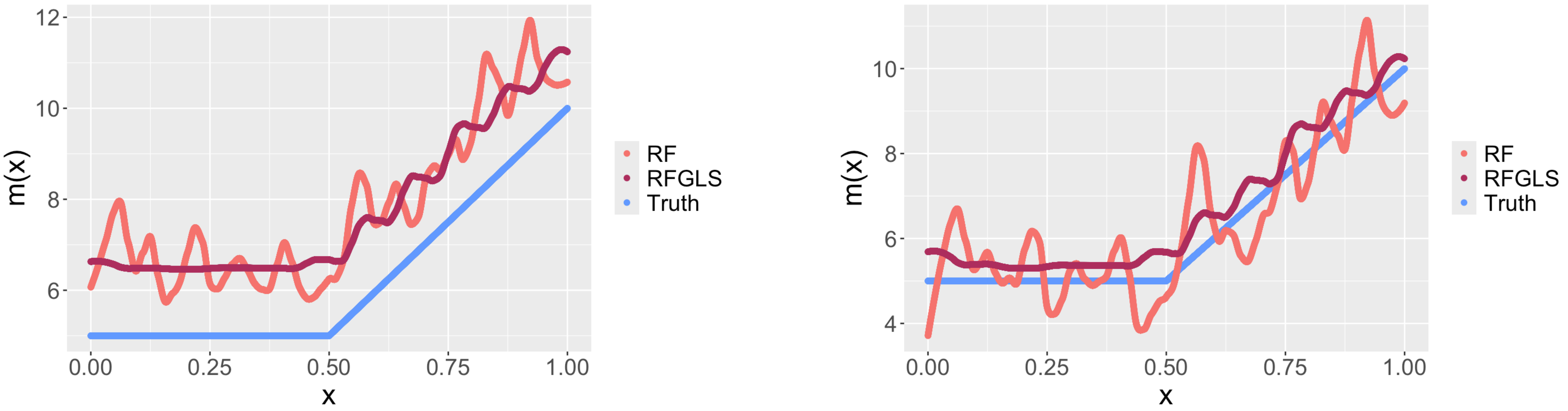


Centered true function and estimates
for data at 500 locations in $[0,1] \times [0,1]$

Mean shift

Mean function estimates can sometimes have a constant shift

- Occurs when locations are densely packed (in-fill sampling)
- Less severe when locations are spread out (increasing domain sampling)

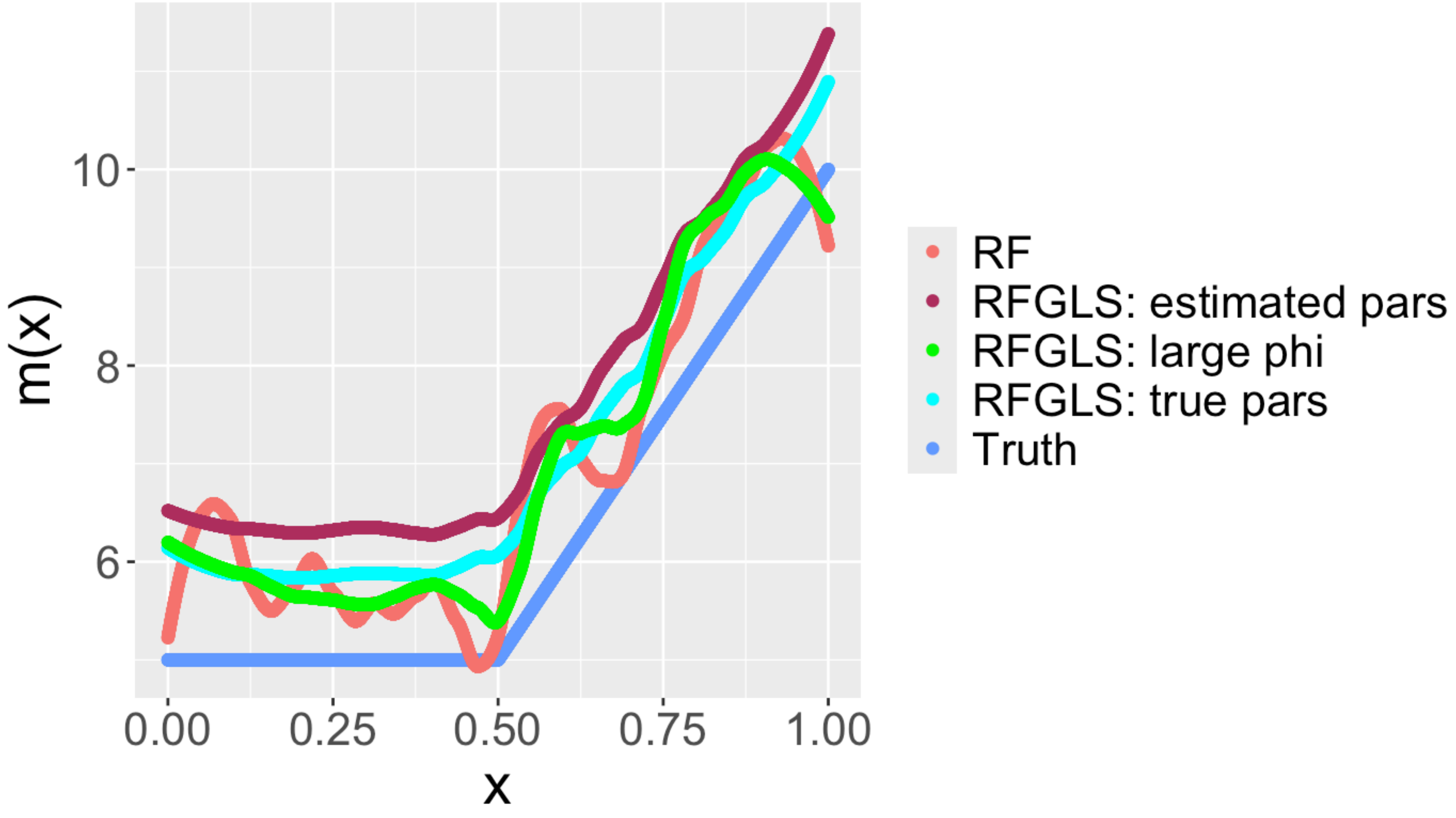


500 locations in $[0,1] \times [0,1]$ 500 locations in $[0,5] \times [0,5]$

Spatial parameter estimation

RFGLS_estimate_spatial can estimate the spatial covariance parameters (set *param_estimate=T*)

It can also use fixed user-input values of these parameters



Spatial parameter estimation

Setting *param_estimate=T* estimates the parameters based on training residuals from an RF fit

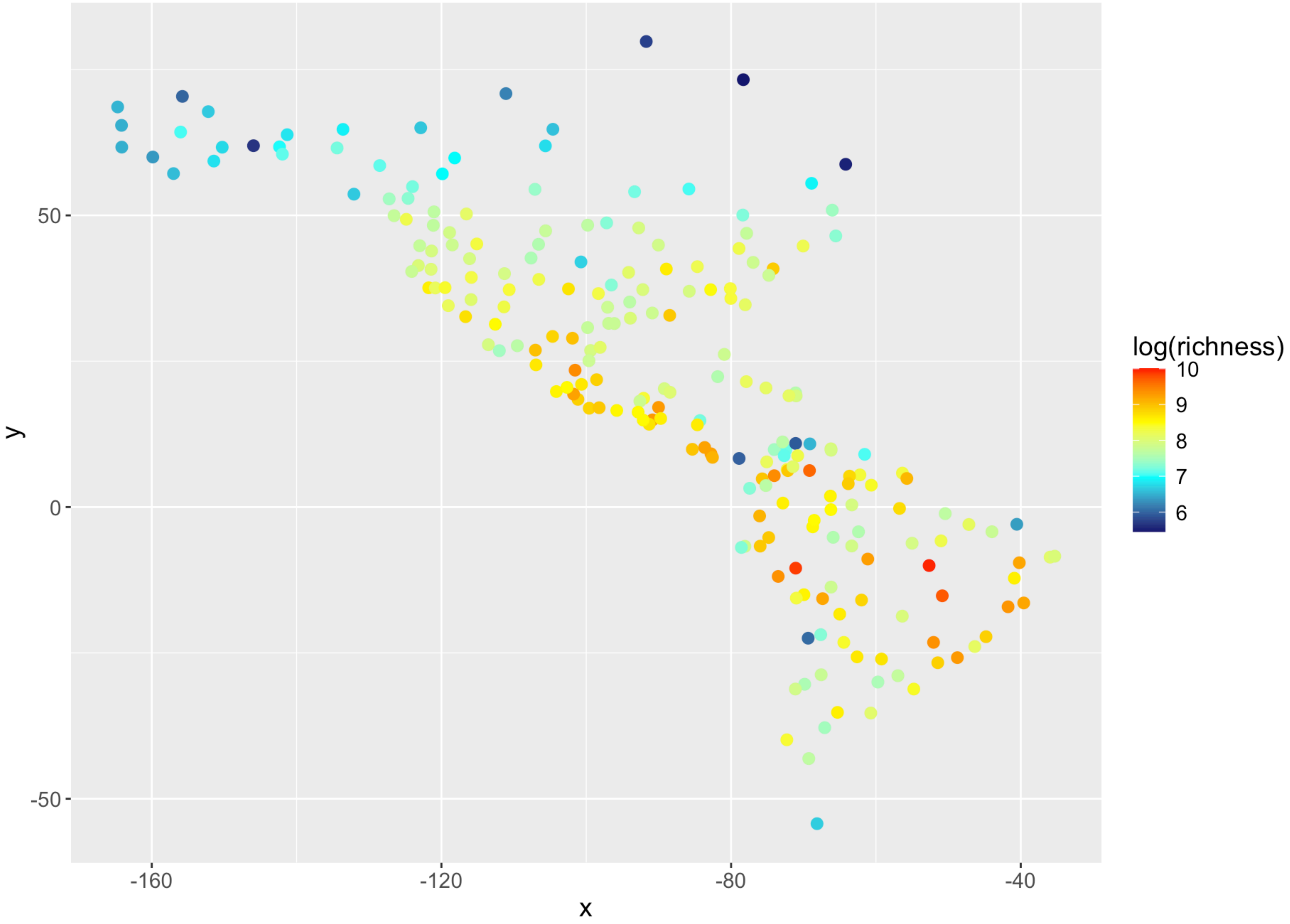
May estimate weaker spatial correlation (small σ^2, large ϕ) if RF overfits

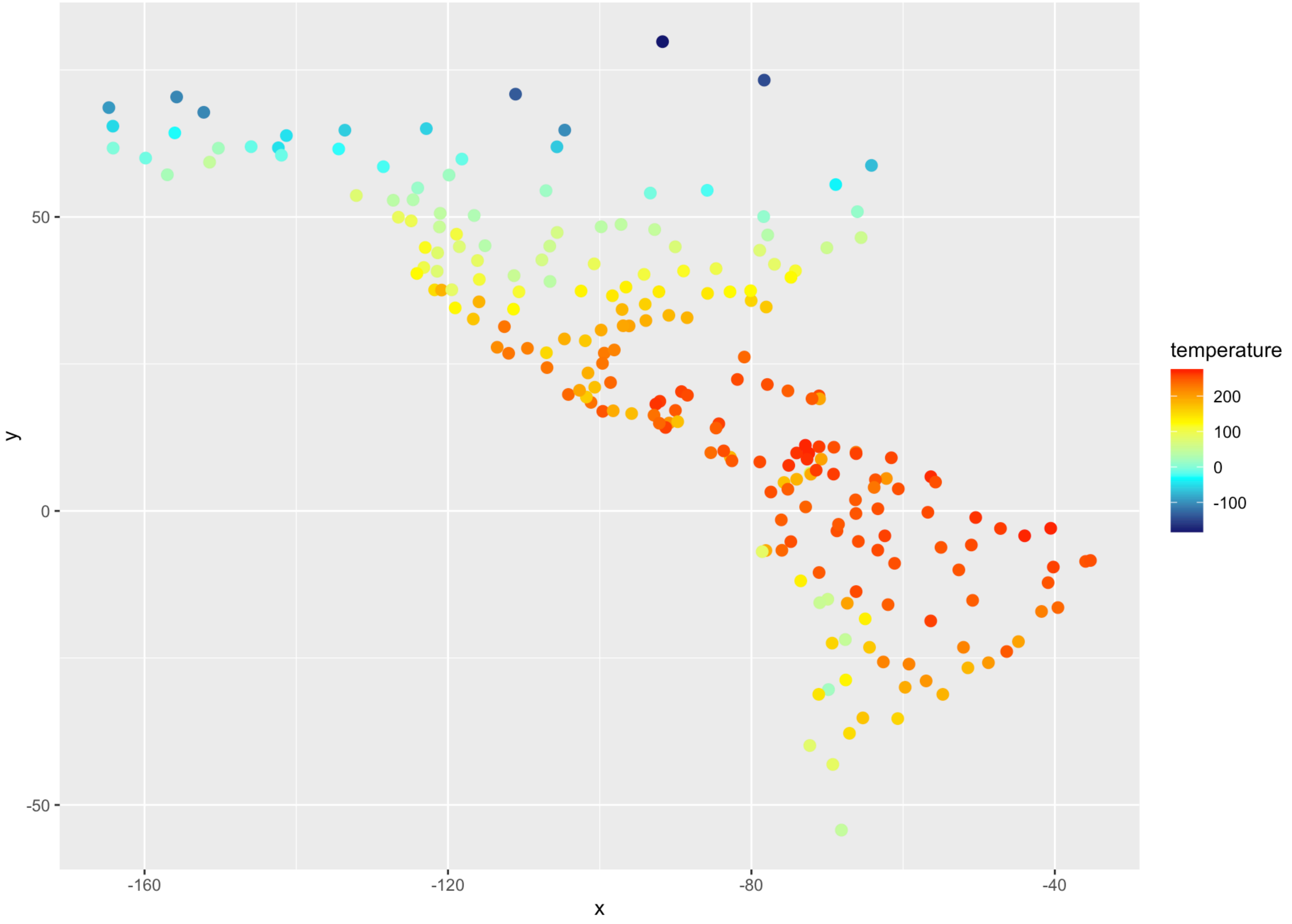
Alternatively, one can use test residuals from an RF-fit to pre-estimate the parameters

Another choice is to use parameter estimates from a spatial linear model (using BRISC)

Plant richness modeling

Dataset on plant richness from the *spatialRF* package

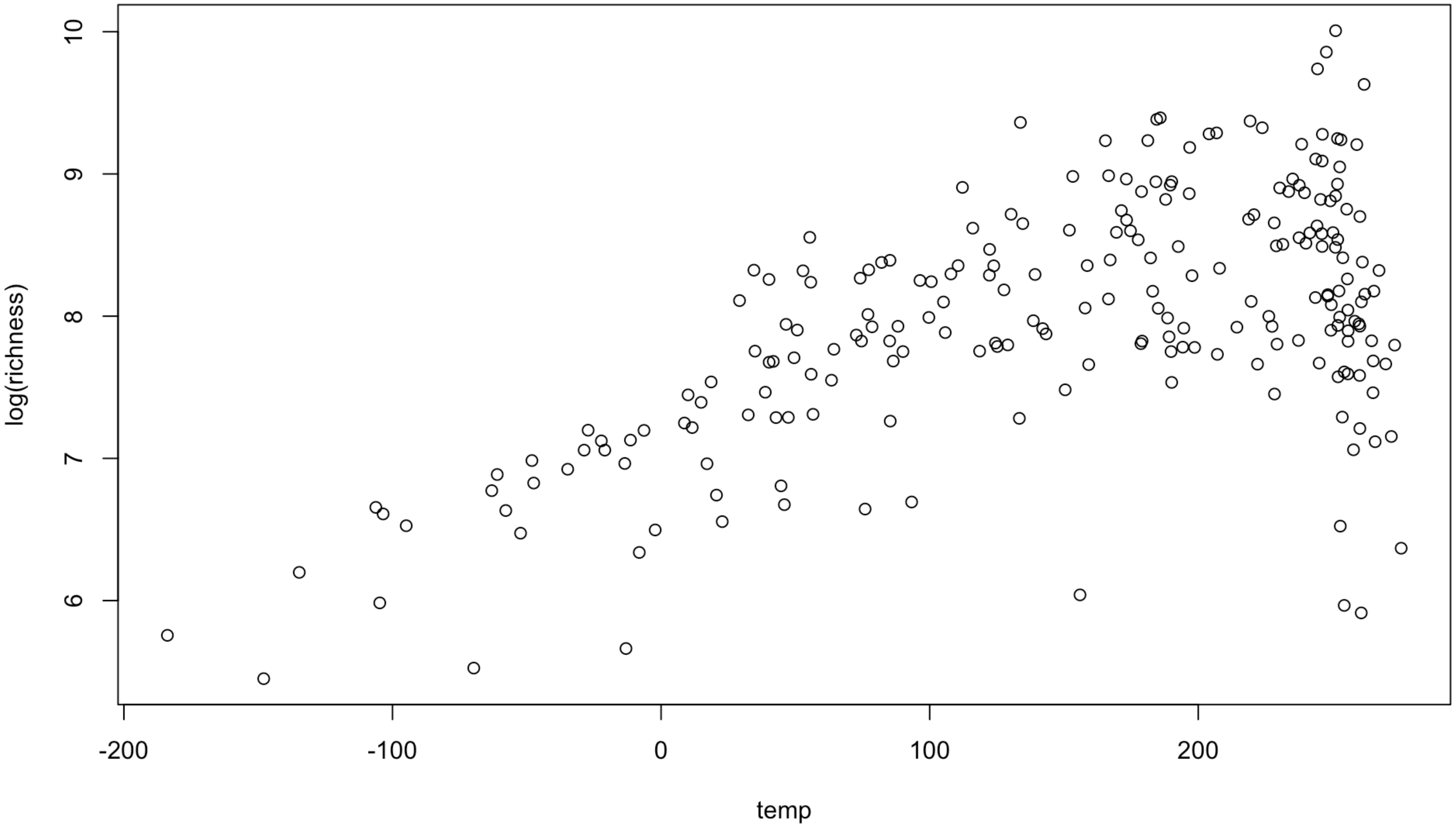




log(rch)=log(richness)

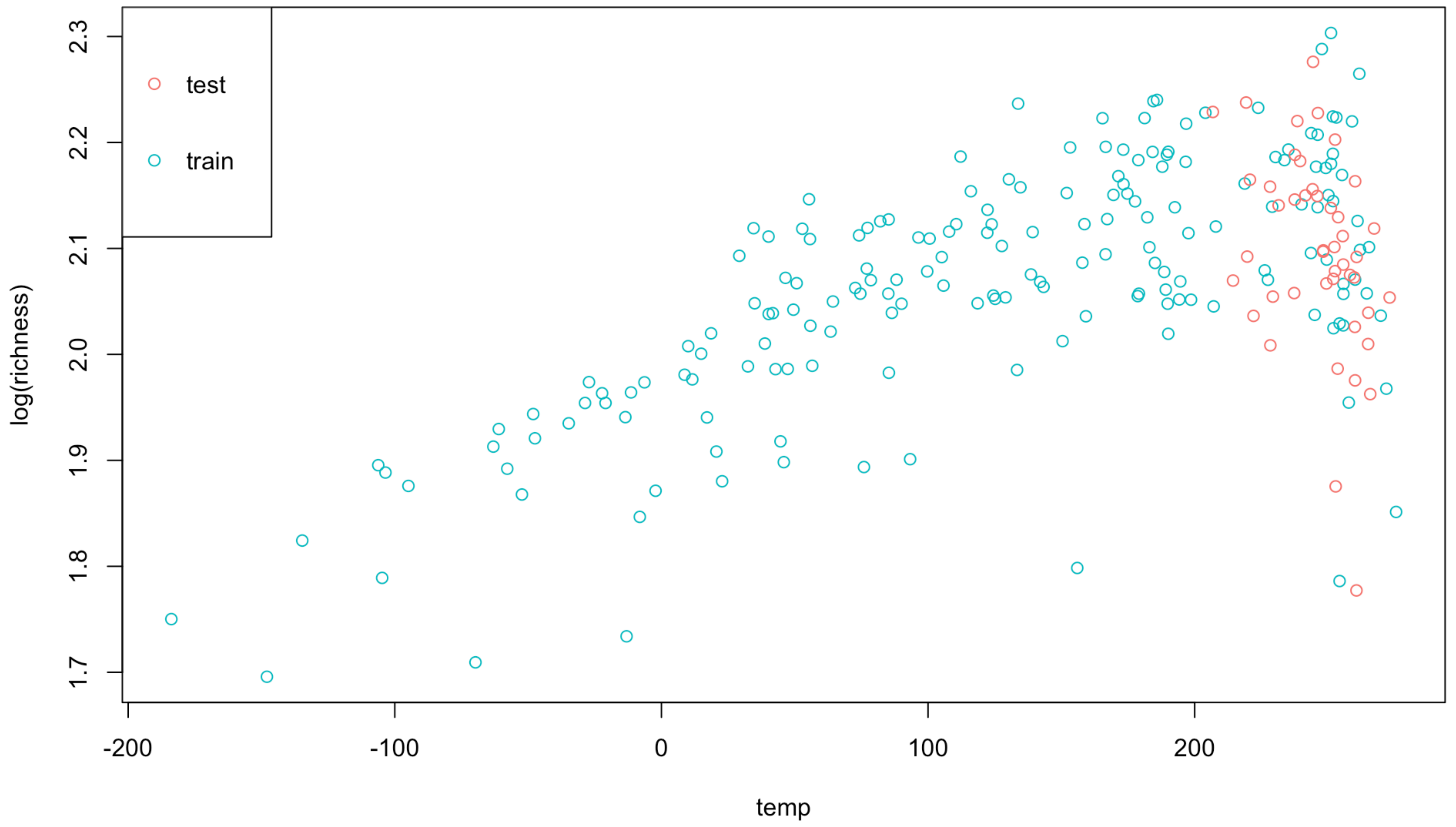
tmp=temperature

Plant richness modeling



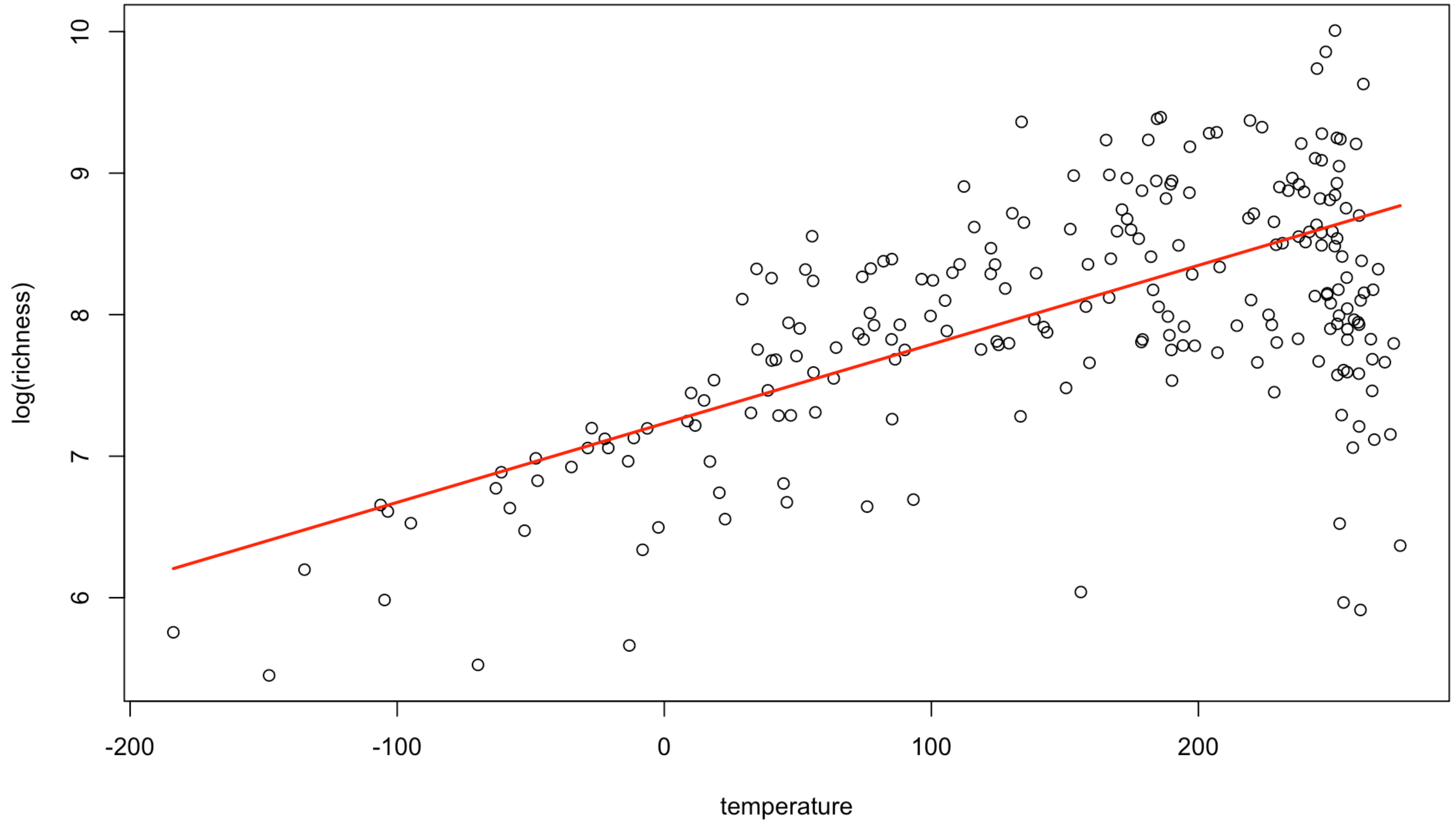
log(rch) vs tmp

Plant richness modeling

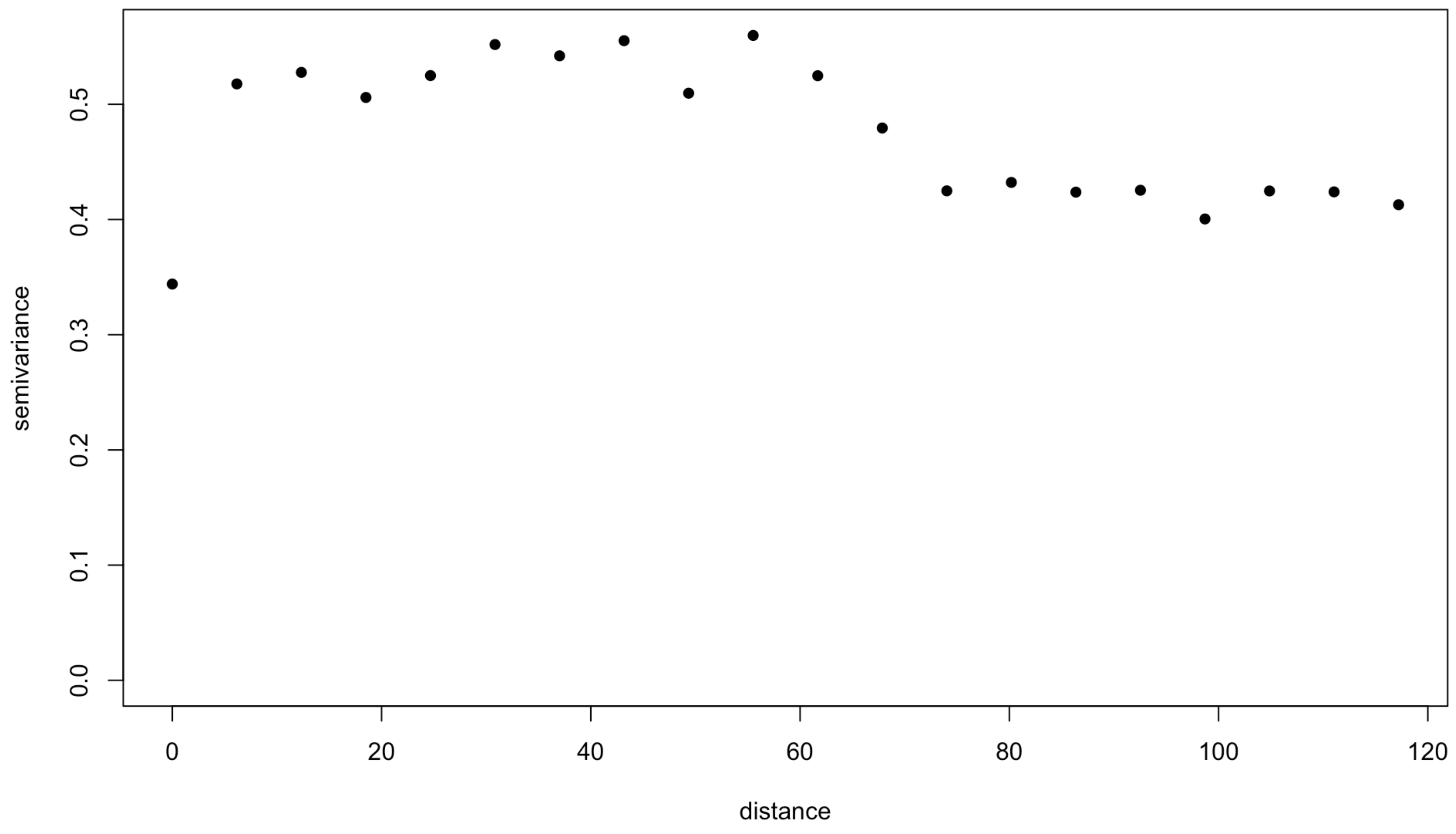


Split of the data into test and train sets

Plant richness modeling



(Non-spatial) linear model fit

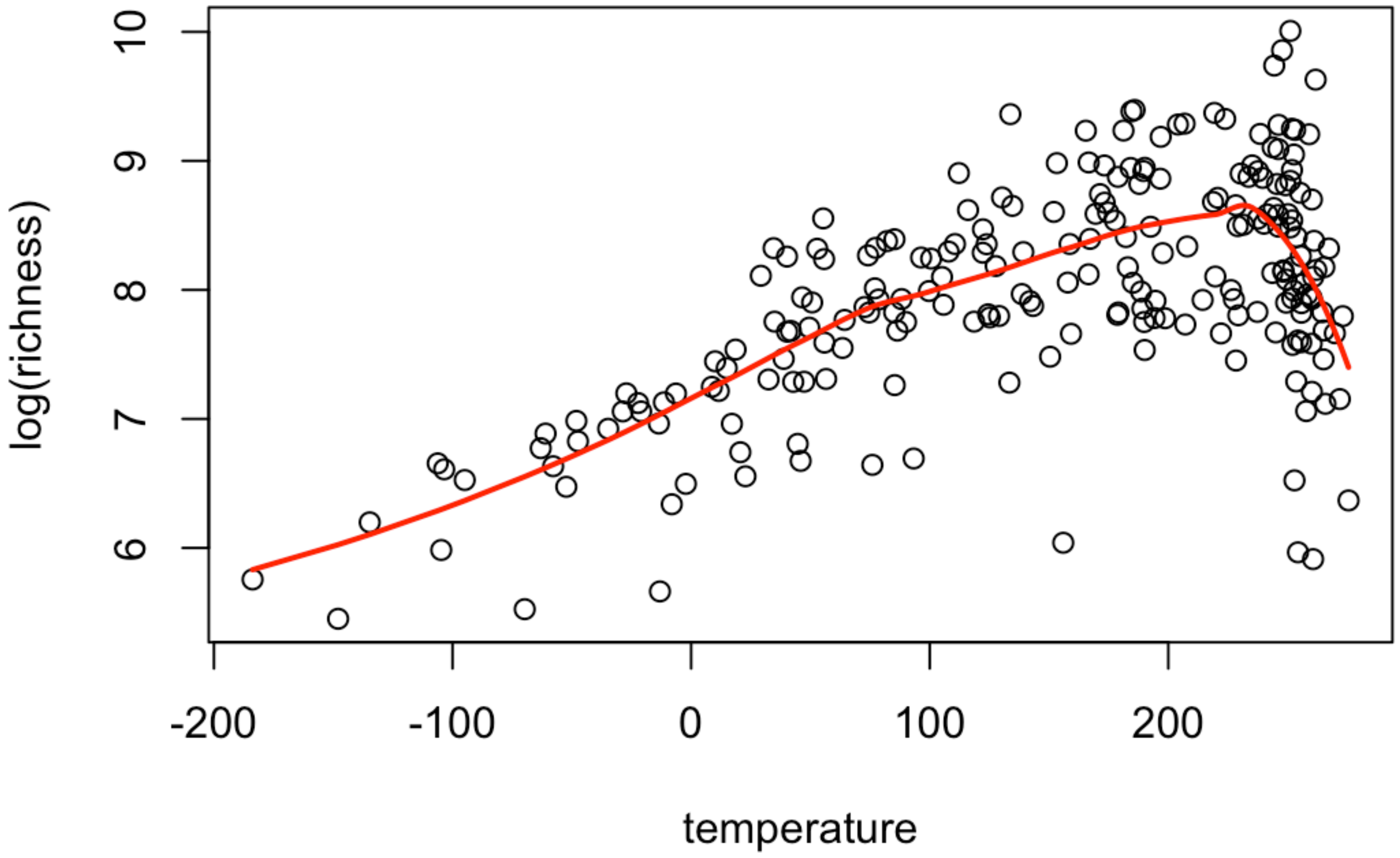


Semi-variogram of the residuals

Plant richness modeling

RMSE for mean estimation $\sum_{i=1}^{n_{test}} (y_i - \hat{m}(X_i))^2$

Method	RMSE
LM	0.89
spLM	0.79
RF	0.84
RFGLS	**0.69**



LM = linear model log(rch) ~ tmp + iid error
spLM = linear model (log(rch) ~ tmp + GP error
RF = random forest log(rch) ~ m(tmp) + iid error
RFGLS = random forest log(rch) ~ m(tmp) + GP error

RFGLS fit of the mean

Plant richness modeling

Methods

LM* = linear model log(rch) ~ tmp + iid error
spLM = linear model log(rch) ~ tmp + GP error
spLM2 = linear model log(rch) ~ tmp + lat + GP error
RF* = random forest log(rch) ~ m(tmp) + iid error
RFGLS = random forest log(rch) ~ m(tmp) + GP error
RFGLS2 = random forest log(rch) ~ m(tmp,lat) + GP error
RF-loc = random forest log(rch) ~ m(tmp,lat,lon) + iid error
spRF = random forest log(rch) ~ m(tmp,pairwise distances) + iid error

* Does not offer spatial predictions, so just the mean predictions are used

RMSPE for spatial predictions

Method	**RMSE**
LM	0.89
spLM	0.71
spLM2	0.70
RF	0.84
RFGLS	0.67
RFGLS2	**0.63**
RFloc	**0.63**
spRF	0.65

RFGLS for time series data

RFGLS can be used to estimate non-linear mean functions in time series data with autoregressive errors

Non-linear AR(q) model: $Y_t = m(X_t) + \epsilon_t, \epsilon_t = \sum_{j=1}^{q} \rho_j \epsilon_{t-j} + \eta_t, \eta_t \sim_{\text{iid}} N(0,\sigma^2)$

RFGLS for time series data

RFGLS can be used to estimate non-linear mean functions in time series data with autoregressive errors

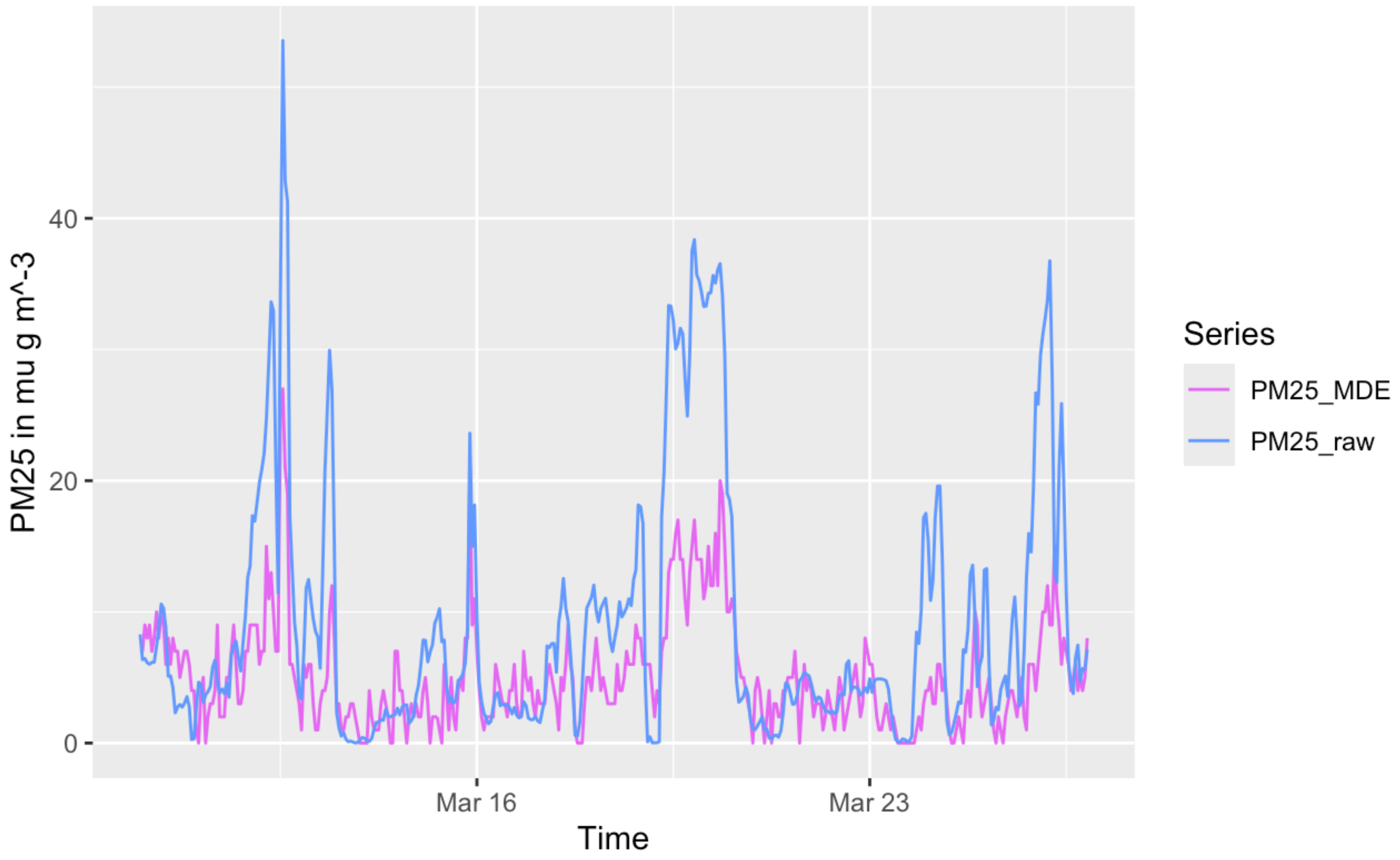
Non-linear AR(q) model: $Y_t = m(X_t) + \epsilon_t, \epsilon_t = \sum_{j=1}^{q} \rho_j \epsilon_{t-j} + \eta_t, \eta_t \sim_{\text{iid}} N(0, \sigma^2)$

Estimation of m using *RFGLS_estimate_timeseries*

- Setting *param_estimate=T* estimates the auto-correlation parameters
- Initial values of auto-correlation parameters can be set using *lag_params*
- If *param_estimate=F,* auto-correlation parameters are fixed at *lag_params* values

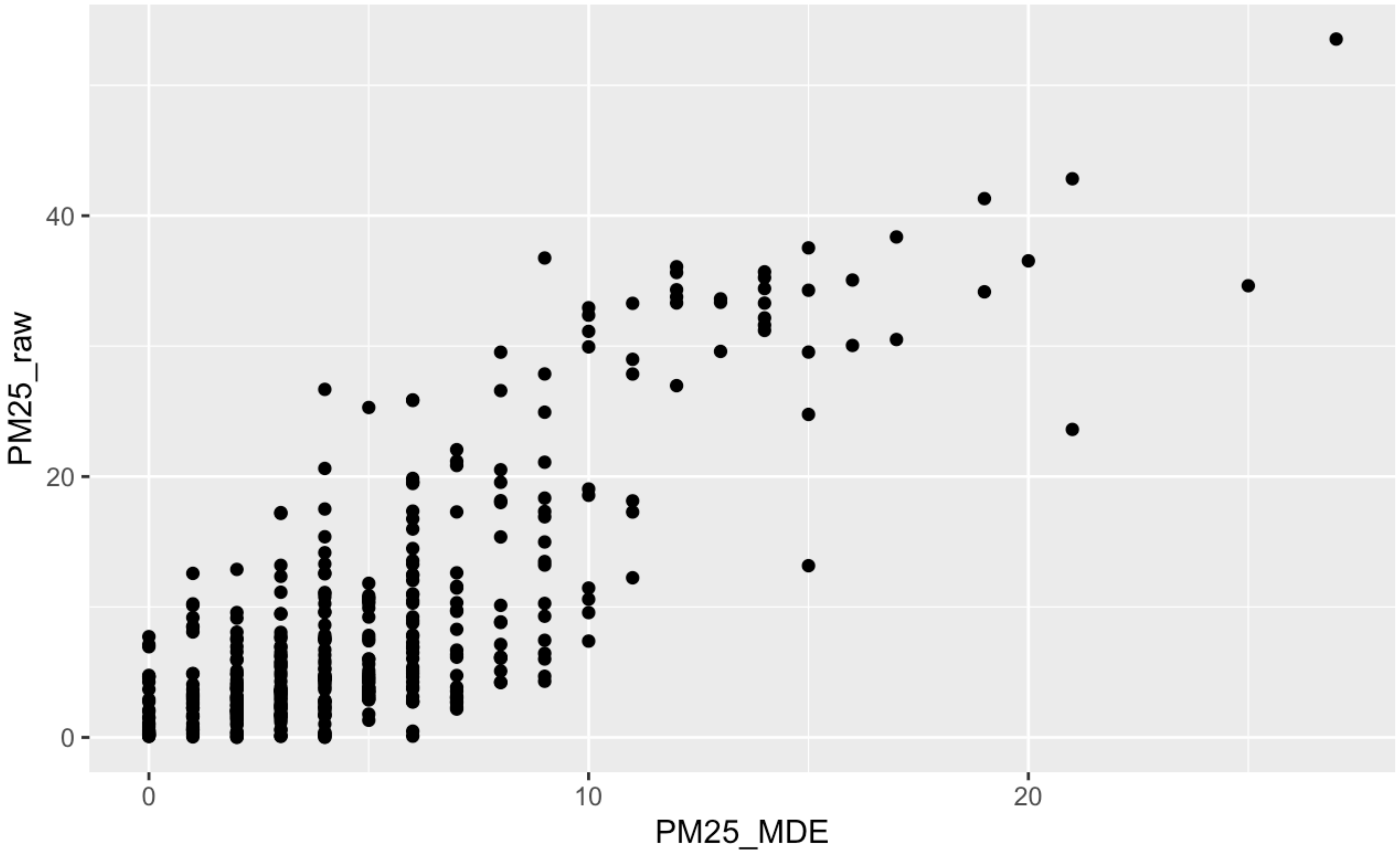
Prediction of m using *RFGLS_predict*

Low-cost sensor air-pollution time-series modeling



Goal: Estimate the mean of the low-cost sensor data (raw) in terms of the higher quality reference data (MDE)

Low-cost sensor air-pollution time-series modeling

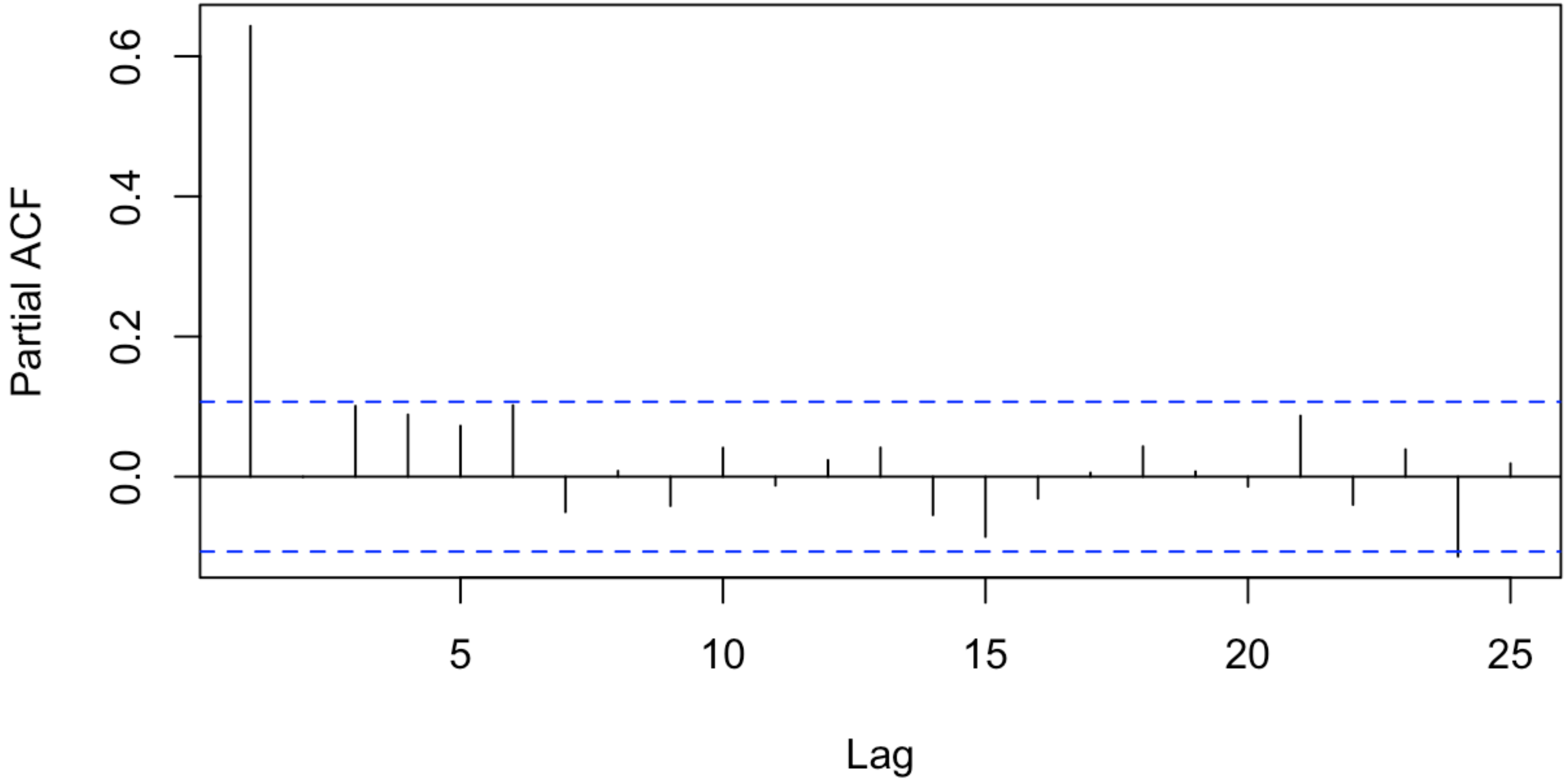


Goal: Estimate the mean of the low-cost sensor data (raw) in terms of the higher quality reference data (MDE)

Low-cost sensor air-pollution time-series modeling

Linear model:

PM25_raw ~ PM25_MDE + iid error



Partial auto-correlation function (*pacf*) plot of the linear model residuals

Low-cost sensor air-pollution time-series modeling

Methods

LM = linear model

PM25_raw ~ PM25_MDE + iid error

RF = random forest

PM25_raw ~ PM25_MDE + iid error

RFGLS = random forest for time series

PM25_raw ~ PM25_MDE + AR(1) error

Method	RMSE
LM	6.9
RF	6.9
RFGLS	**6.4**

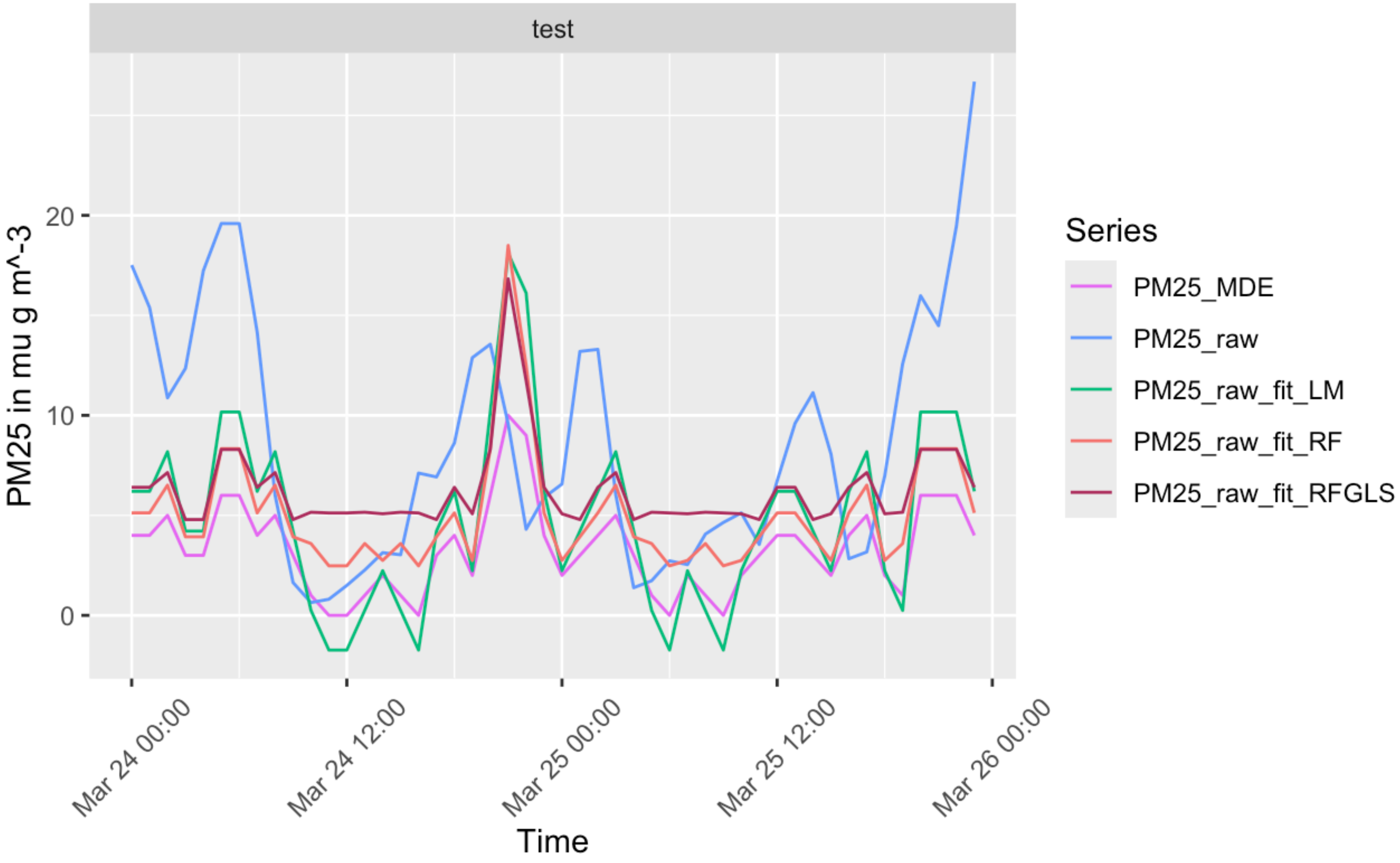
Train:

Hourly data from days 1 to 12 ($n = 24 * 12 = 288$)

Test:

Hourly data from days 15 and 16

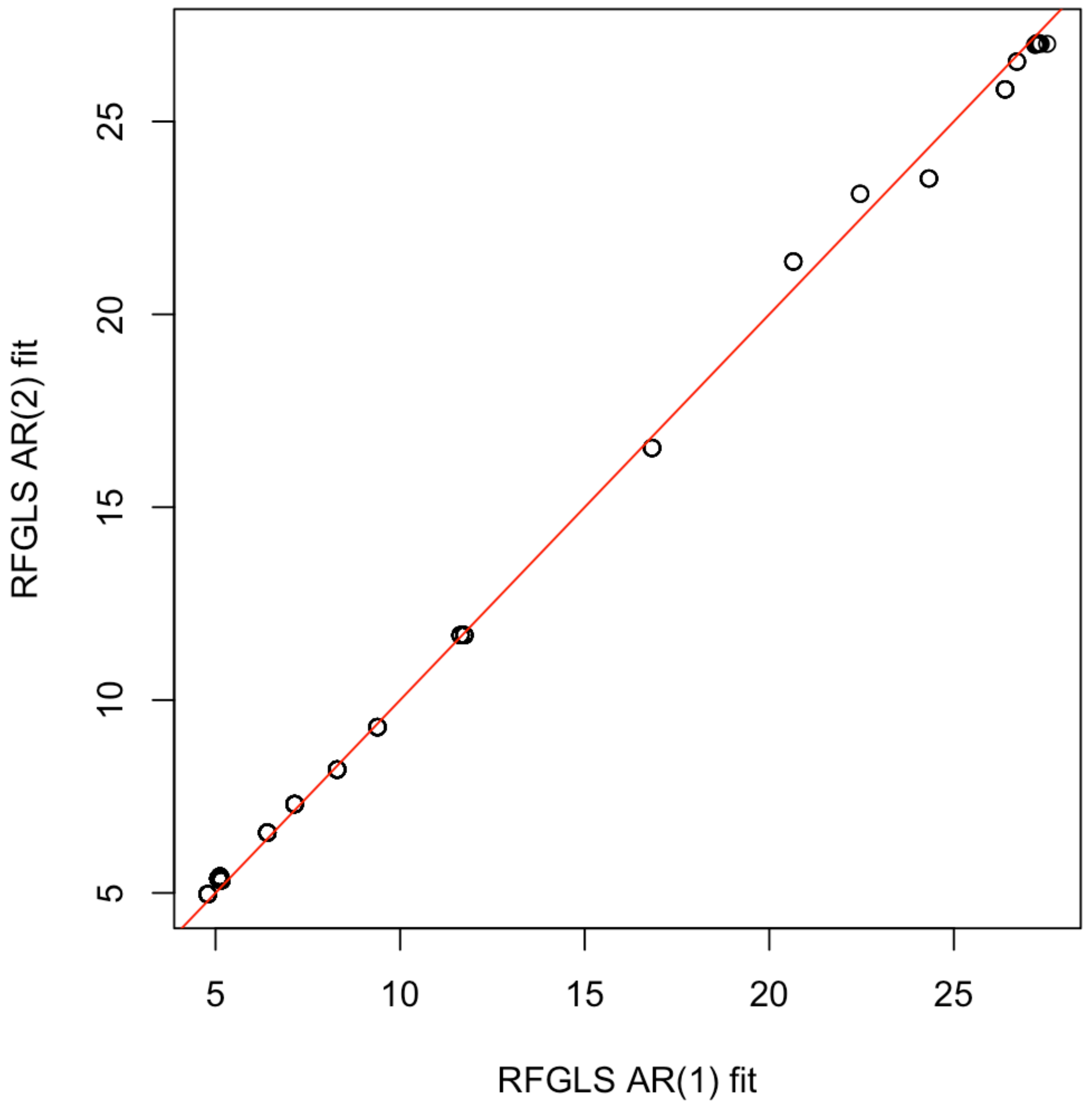
Low-cost sensor air-pollution time-series modeling



Low-cost sensor air-pollution time-series modeling

Order of autoregression: RFGLS can use higher order AR models

To fit AR model of order q, simply provide a q-dimensional vector input value of *lag_params*



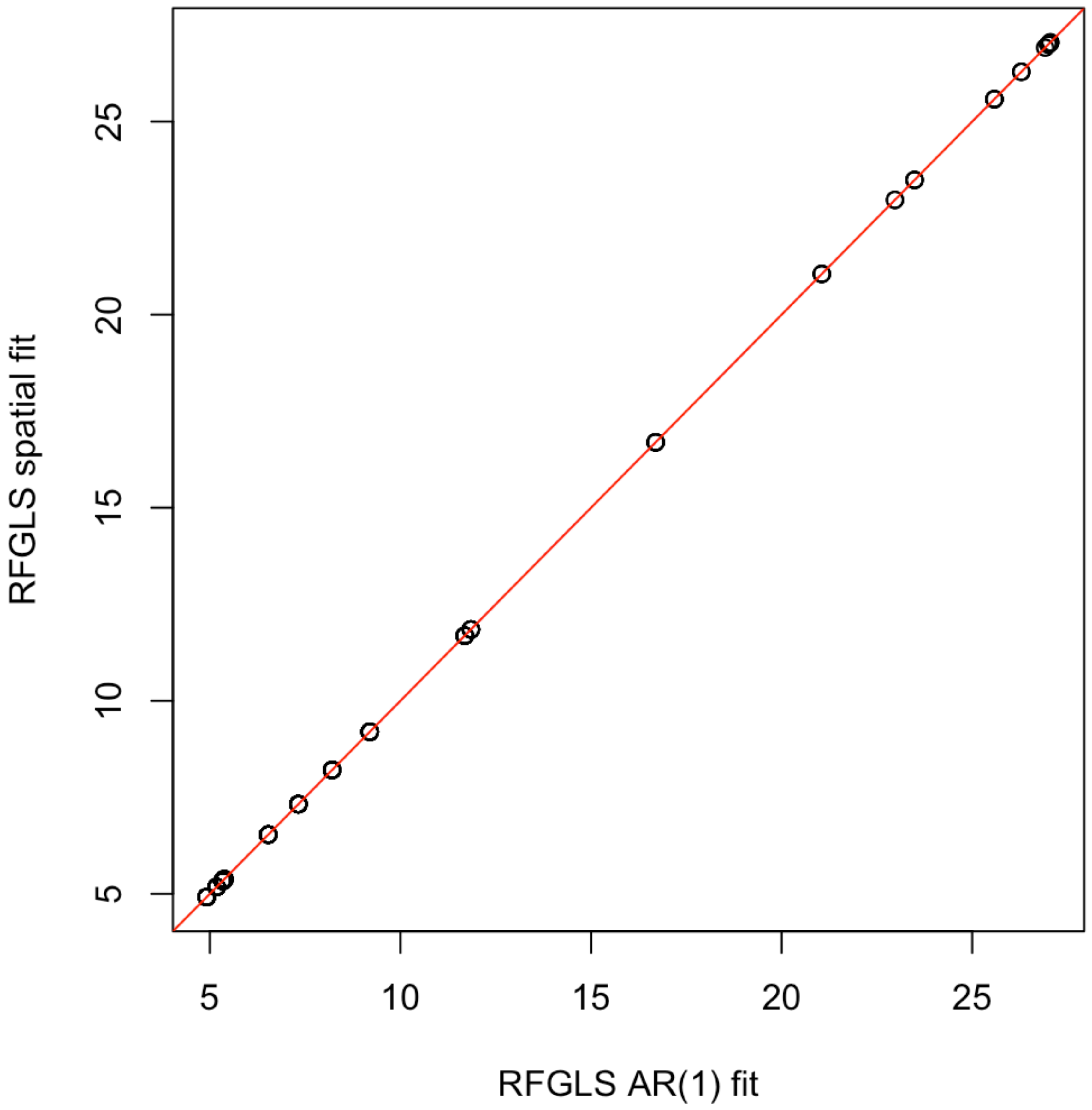
Fits from RFGLS with AR(1) error vs RFGLS with AR(2) error for the PM25 data

Low-cost sensor air-pollution time-series modeling

RFGLS_estimate_timeseries only works for datasets with equi-spaced time-points

Unequally spaced time-series data can be analyzed by *RFGLS_estimate_spatial*

- Treats time as 1-dimensional space
- Leverages equivalence of AR(1) and exponential GP covariance matrices



Fits from RFGLS with AR(1) error vs RFGLS with exponential GP error for the PM25 data

Asymptotic Theory for RFGLS

If the errors are sub-Gaussian stationary $\beta-$mixing (absolutely regular) process, then under regularity conditions on the working precision matrix, RF-GLS is consistent for m.

Examples where the consistency holds:

- Spatial Matérn GP on 1-dimensional lattice
- Autoregressive time-series

To our knowledge, first theory of random forests for spatially dependent data

Summary

The linearity assumption of spatial mixed effect models can sometimes be inadequate

Non-linear machine learning methods like random forests and neural networks are being increasingly adopted for geospatial analysis

- Cannot directly model spatial correlation as done in mixed models via Gaussian Process errors
- Spatial correlation is often ignored for mean function estimation using random forests
- Latitude-longitude or pairwise distances used as additional features in random forests can only offer prediction

Summary

RF-GLS: A model based framework, embedding RF within the spatial mixed models

- Spatial correlation directly modeled using Gaussian process errors
- Non-linear mean function estimated using random forests by accounting for spatial correlation (DART loss and GLS style trees)
- Spatially-informed predictions using GP via kriging

RFGLS can also be used non-linear trend (mean) estimation in time series data

Asymptotic theory of RFGLS for dependent data

RandomForestsGLS R-package

- Estimation and prediction using RFGLS in spatial and time-series data
- Computational strategies using rounding, binning, parallelization

Main References

RFGLS paper: Saha, A., Basu, S., & Datta, A. (2023). *Random forests for spatially dependent data*. Journal of the American Statistical Association, 118(541), 665-683.

RFGLS software paper: Saha, A., Basu, S., & Datta, A. (2022). *RandomForestsGLS: an r package for random forests for dependent data*. Journal of open source software, 7(71)

RFGLS software: https://cran.r-project.org/web/packages/RandomForestsGLS/

Other references

Hornik, K., Stinchcombe, M., & White, H. (1989). *Multilayer feedforward networks are universal approximators.* Neural networks, 2(5), 359-366.

Diggle, P. J., & Hutchinson, M. F. (1989). *On spline smoothing with autocorrelated errors.* Australian Journal of Statistics, 31(1), 166-182.

Breiman, L. (2001). *Random forests.* Machine learning, 45, 5-32.

Taylor, J., & Einbeck, J. (2013). *Challenging the curse of dimensionality in multivariate local linear regression.* Computational Statistics, 28, 955-976.

Scornet, E. (2016). *On the asymptotics of random forests.* Journal of Multivariate Analysis, 146, 72-83.

Fayad, I., Baghdadi, N., Guitet, S., Bailly, J. S., Hérault, B., Gond, V., ... & Minh, D. H. T. (2016). *Above ground biomass mapping in French Guiana by combining remote sensing, forest inventories and environmental data*. International Journal of Applied Earth Observation and Geoinformation, 52, 502-514.

Nandy, S., Lim, C. Y., & Maiti, T. (2017). *Additive model building for spatial regression*. Journal of the Royal Statistical Society Series B: Statistical Methodology, 79(3), 779-800.

Hengl, T., Nussbaum, M., Wright, M. N., Heuvelink, G. B., & Gräler, B. (2018). *Random forest as a generic framework for predictive modeling of spatial and spatio-temporal variables*. PeerJ, 6, e5518.

Fox, E. W., Ver Hoef, J. M., & Olsen, A. R. (2020). *Comparing spatial regression to random forests for large environmental data sets*. PloS one, 15(3), e0229509.

Schmidt-Hieber, J. (2020). *Nonparametric regression using deep neural networks with ReLU activation function*, Annals of Statistics 1875-1897.