TBASS: A Robust Adaptation of Bayesian Adaptive Spline Surfaces

Andy A. Shen, Kellin N. Rumsey, Devin C. Francom

Statistical Sciences Group (CCS-6): Los Alamos National Laboratory

Abstract

The R package 'TBASS' is an extension of the 'BASS' package created by Francom et. al (2016). The package is used to fit a Bayesian adaptive spline surface to a dataset that follows a Student's t-distribution or has outliers. Much of the framework for 'TBASS' is adapted from the concepts of Bayesian Multivariate Adaptive Regression Splines (BMARS), specifically the work done from Denison, Mallick, and Smith (1998). By including a more robust generalization, a dataset with outliers can now be accurately fit using the BMARS model, without the possibility of overfitting or variance inflation.

Keywords: splines, robust regression, Bayesian inference, nonparametric regression, sensitivity analysis

1 Introduction

Splines are a commonly used regression tool for fitting nonlinear data, univariate and multivariate. Splines can act as basis functions, where all of the basis functions combine to form the $\boldsymbol{X}$ matrix. The simplest way to create the ith basis functions can be represented as

$$X_{ij} = [s_i(x_j - t_i)]_+ \tag{1}$$

Equation (1) is used to calculate the ith column of the X matrix of basis functions, where $s_i \in -1, 1$, t_i is called a **knot** and $[a]_+ = max(0, a)$.

For example, given the nonlinear data shown in Figure 1 below, we can use (1) to fit a spline model shown in Figure 2.

Figure 1 : Univariate Nonlinear Data

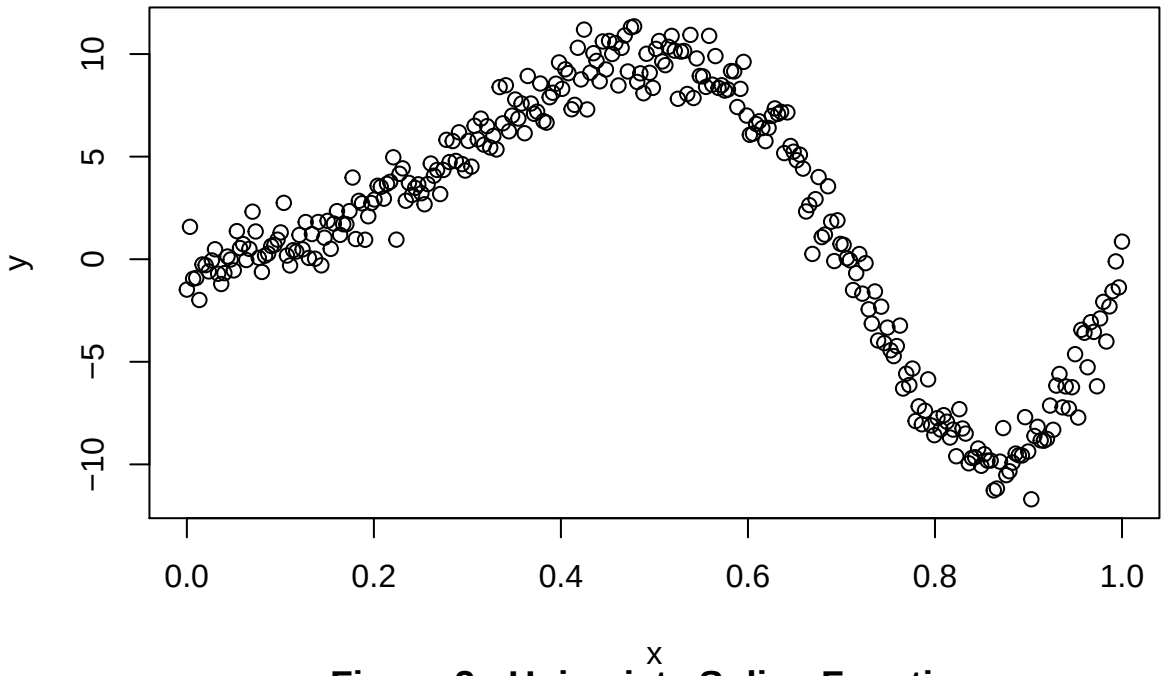
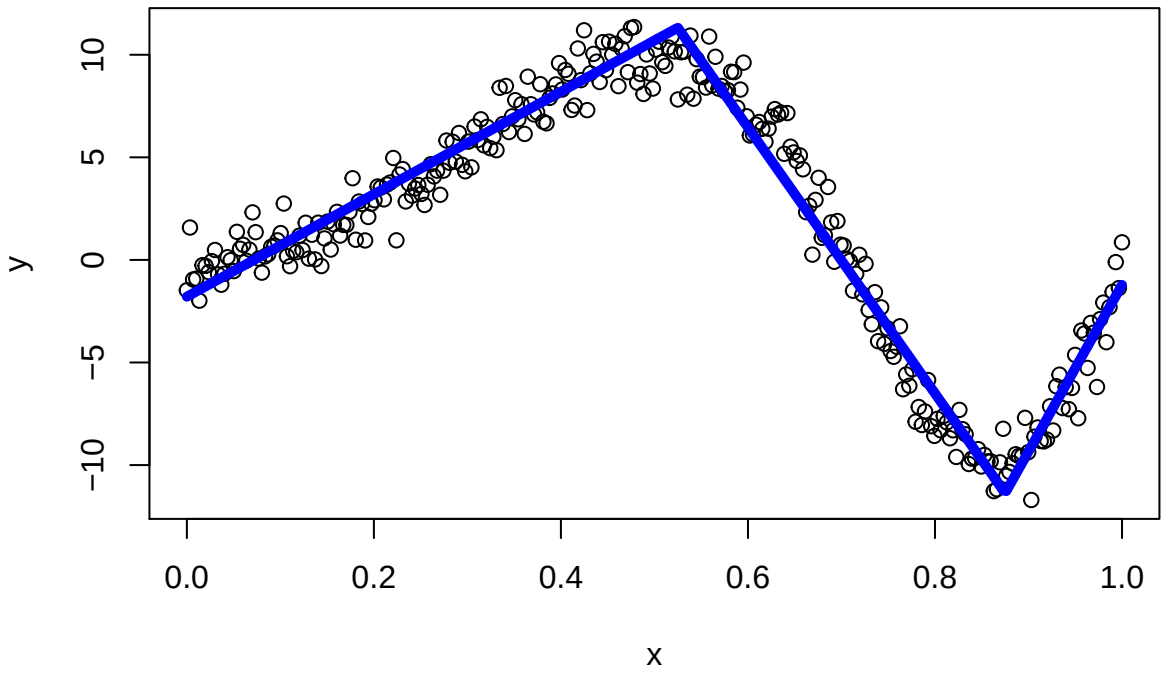


Figure 2 : Univariate Spline Function



2 Robust BMARS

2.1 Overview

We want to extend the theory behind frequentist univariate spline regression to a multivariate Bayesian framework. Moreover, we want to be able to fit nonlinear data that has outliers. We adopt the Gaussian BMARS framework based on the standard framework used by Dennison et al. (1998) and Francom (2018) when deriving our likelihood and full conditional distributions for the parameters.

In the presence of outliers, Gaussian BMARS will attempt to capture the excess noise by adding basis functions (overfitting) or inflating the variance term (σ^2). The Robust BMARS model accounts for this sensitivity to outliers by avoiding overfitting or variance inflation when the degrees of freedom (ν) are low. When ν is high, the t-distribution closely mimics a normal distribution, so the Robust BMARS model behaves in a similar way.

The function used to fit the Robust BMARS model in the `TBASS` package is the `tbass()` command (see section 3).

2.2 Auxiliary Variables

In creating the Robust BMARS model, we introduce new auxiliary parameters based on the work done by Gelman et al. (2014).

Similar to Gaussian BMARS, we let y_i be our dependent variable and $\mathbf{x}_i$ be our independent variable representing a single basis function with $i = 1, ..., n$. Without loss of generality, all independent variables $\mathbf{x}_i$ are scaled from zero to one (Francom, 2016).

In the robust case, y_i is modeled as

$$\boldsymbol{y} = X\boldsymbol{\beta} + \epsilon, \quad \epsilon \sim t_\nu \left(0,\ \sigma^2 V^{-1}\right) \tag{2}$$

and

$$y_i | V_i \sim \mathcal{N}\left(X\boldsymbol{\beta},\ \frac{\sigma^2}{V_i}\right) \tag{3}$$

where $\boldsymbol{X}$ represents the matrix of basis functions by column, $\boldsymbol{\beta}$ is the vector of regression coefficients, ϵ is the error term, ν represents the degrees of freedom in a Student's t-distribubtion, σ^2 is the variance term for y_i, and $V^{-1} = diag(1/V_1 ... 1/V_n)$ where V_i is the variance estimate of y_i.

The basis functions themselves are produced the same way as in the `BASS` package by Francom, et al. (2016).

2.3 Priors

We assume an Inverse-Gamma prior for σ^2 with default shape $\gamma_1 = 0$ and default rate $\gamma_2 = 0$, and a Gamma prior with shape and rate $\frac{\nu}{2}$ for V_i, such that

$$\sigma^2 \sim IG\ (\gamma_1,\ \gamma_2) \tag{4}$$

$$V_i \sim \Gamma\left(\frac{\nu}{2},\ \frac{\nu}{2}\right) \tag{5}$$

From there, we obtain the full conditional of V_i as

$$V_i|\cdot \sim \Gamma\left\{\frac{\nu+1}{2},\ \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y-X\beta)^2\right\} \tag{6}$$

For the parameter governing the number of basis functions λ, we have that

$$\lambda|\cdot \sim \Gamma\left(h_1+M,\ h_2+1\right) \tag{7}$$

where $h_1 = h_2 = 10$ are the default hyperparameters for λ and M is the current number of basis functions. It follows that λ follows a gamma prior.

2.4 Regression Coefficients

Finally, our regression coefficients $\boldsymbol{\beta}$ follow a Gaussian prior such that

$$\boldsymbol{\beta}|\cdot \sim \mathcal{N}\left(0,\ \tau^2 I\right) \tag{8}$$

In the Student's t-distribution, we can marginalize the posterior for $\boldsymbol{\beta}$ and σ^2 to obtain the regression estimate $\hat{\boldsymbol{\beta}}$:

$$\hat{\boldsymbol{\beta}} = \frac{1}{\sigma^2}\left(\frac{1}{\sigma^2}X'VX + \tau^{-2}I\right)^{-1}X'Vy \tag{9}$$

where τ^2 is the prior variance for β_i.

2.5 Likelihood Function

Estimate (9) allows us to achieve our t-distributed likelihood function for the birth step. After marginalizing out $\boldsymbol{\beta}$ and σ^2, we can simplify our Likelihood L to

$$L \propto \left(\tau^2\right)^{-\frac{M+1}{2}}\ |V|^{-1/2}\ \left|H^{-1}\right|^{-1/2} exp\left(-\frac{1}{2}y'V^{-1}y\right) exp\left(-\frac{1}{2}\boldsymbol{\beta}'H^{-1}\boldsymbol{\beta} - 2\boldsymbol{\beta}'X'V^{-1}y\right)$$

where $H = (X'V^{-1}X + \tau^{-2}I)$.

We then complete the square on the first exponential term using $\$y`V^{-1}X H^{-1}V`y \$ which simplifies to $\hat{\boldsymbol{\beta}}' H\hat{\boldsymbol{\beta}}$ since $X'V^{-1}y = H\hat{\beta}$ from (9).

From there, we have that

$$
\begin{aligned}
L(M) &\propto \left(2\pi\tau^2\right)^{-\frac{M+1}{2}} \ |V|^{-1/2} \ exp\left(-\frac{1}{2}\left(y'V^{-1}y - \hat{\boldsymbol{\beta}}'H\hat{\boldsymbol{\beta}}\right)\right) exp\left(-\frac{1}{2}\left(\boldsymbol{\beta}'H\boldsymbol{\beta} - \boldsymbol{\beta}'H\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}'H\boldsymbol{\beta} + \hat{\boldsymbol{\beta}}'H\hat{\boldsymbol{\beta}}\right)\right) \\
&\propto \left(2\pi\tau^2\right)^{-\frac{M+1}{2}} \ |V|^{-1/2} \ exp\left(-\frac{1}{2}\left(y'V^{-1}y - \hat{\boldsymbol{\beta}}'H\hat{\boldsymbol{\beta}}\right)\right) exp\left\{-\frac{1}{2}\left[\left(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}\right)' H \left(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}\right)\right]\right\} \\
&\propto \left(2\pi\tau^2\right)^{-\frac{M+1}{2}} \ |V|^{-1/2} \ exp\left(-\frac{1}{2}\left(y'V^{-1}y - \hat{\boldsymbol{\beta}}'H\hat{\boldsymbol{\beta}}\right)\right) \ \mathcal{N}\left(\boldsymbol{\beta} \mid \hat{\boldsymbol{\beta}}, \ H^{-1}\right) \left|H^{-1}\right|^{-1/2} (2\pi)^{\frac{M+1}{2}} \\
&\propto \left(\tau^2\right)^{-\frac{M+1}{2}} |V|^{-1/2} \ \left|H^{-1}\right|^{-1/2} exp\left\{-\frac{1}{2}\left(y'V^{-1}y - \hat{\boldsymbol{\beta}}'H\hat{\boldsymbol{\beta}}\right)\right\} \\
&\propto \left(\tau^2\right)^{-\frac{M+1}{2}} |V|^{-1/2} \ \left|H^{-1}\right|^{-1/2} exp\left\{-\frac{1}{2}\left(y'V^{-1}y - \hat{\boldsymbol{\beta}}'\left(X'V^{-1}X + \tau^{-2}I\right)\hat{\boldsymbol{\beta}}\right)\right\}
\end{aligned}
$$

$$
\Rightarrow L(M) = (\tau^2)^{\frac{M+1}{2}} |V|^{-1/2} \ \left|(X^tV^{-1}X + \tau^{-2}I)^{-1}\right|^{-1/2} exp\left\{ -\frac{1}{2}(y'V^{-1}y - \hat{\boldsymbol{\beta}}^t(X^tV^{-1}X + \tau^{-2}I)^{-1}\hat{\boldsymbol{\beta}})\right\} \tag{10}
$$

Equation (10) is the likelihood for the current state model with M basis functions.

2.6 Reversible-Jump Markov Chain Monte Carlo (RJ-MCMC)

Like Gaussian BMARS, the robust algorithm builds basis functions adaptively, sampling from candidate knot locations, signs, interaction degrees, and accepting or rejecting the basis functions using a RJ-MCMC algorithm to sample from the full posterior.

RJ-MCMC is an generalization of the traditional Metropolis-Hastings algorithm in the sense that RJ-MCMC allows for parameter dimension change, allowing for simulation when the number of parameters is unknown. This is important for BMARS because we want to learn where the knots should be placed and how many basis functions to have in our model, along with the degree of interaction for our basis functions. There can exist multiple basis functions in a multivariate setting. We also want to know if certain basis functions should be added, deleted, or changed.

The BMARS model has three possible move types, which are sampled using a discrete uniform:

- **Birth**: adding a basis function
- **Death**: deleting a basis function
- **Change**: changing a knot, sign, and values of a basis function

Once the move type is sampled, the RJ-MCMC algorithm is used to determine acceptance of that move type.

Our acceptance ratio α is denoted by

$$
\alpha = min\left\{1, \ \frac{L(D|\theta') \ p(\theta') \ S(\theta' \to \theta)}{L(D|\theta) \ p(\theta) \ S(\theta \to \theta')}\right\} \tag{11}
$$

where θ' represents the candidate model parameters and θ represent the current state model parameters, L is the Gaussian likelihood, p is the prior, and S is the proposal to jump from one model to another.

Section 2.7 details the RJ-MCMC algorithm for the birth step in detail, and the death and change steps are very similar.

2.7 Birth Step

Since a basis function is added for the birth step, we have that

$$L(M+1) \propto \left(\tau^2\right)^{-\frac{M+2}{2}} |V|^{-1/2} \ \left|H^{-1}\right|^{-1/2} exp\left\{-\frac{1}{2}\left(y'V^{-1}y - \hat{\boldsymbol{\beta}}'\left(X'V^{-1}X + \tau^{-2}I\right)\hat{\boldsymbol{\beta}}\right)\right\} \tag{12}$$

Using (10), The ratio $\frac{L(M+1)}{L(M)}$ is then represented as

$$\frac{L(M+1)}{L(M)} = \frac{\left(\tau^2\right)^{-\frac{M+2}{2}} |V|^{-1/2} \ \left|H_c^{-1}\right|^{-1/2} exp\left\{-\frac{1}{2}\left(y'V^{-1}y - \hat{\boldsymbol{\beta}}_c'\left(X_c'V^{-1}X_c + \tau^{-2}I_c\right)\hat{\boldsymbol{\beta}}_c\right)\right\}}{\left(\tau^2\right)^{-\frac{M+1}{2}} |V|^{-1/2} \ |H^{-1}|^{-1/2} exp\left\{-\frac{1}{2}\left(y'V^{-1}y - \hat{\boldsymbol{\beta}}'\left(X'V^{-1}X + \tau^{-2}I\right)\hat{\boldsymbol{\beta}}\right)\right\}} \tag{13}$$

where the subscript c denotes the candidate move type of the algorithm (a birth step in this case).

The death is very similar to the birth step, with a candidate likelihood of $L(M-1)$, and there is no dimension change in the change step.

The RJ-MCMC algorithm calculates the likelihoods, priors, proposals, and acceptance ratios all on the log scale.

2.8 Gibbs Sampling

Once the basis function move type is complete, the model parameter values λ, $\boldsymbol{\beta}$, V_i and σ^2 can then be sampled using Gibbs Sampling, since the full conditionals are all closed-form.

The full conditionals are of the following form:

$$\lambda|\cdot \sim \Gamma\big(h_1 + M,\ h_2 + 1\big) \tag{14}$$

$$\boldsymbol{\beta}|\cdot \sim \mathcal{N}\Big(\hat{\boldsymbol{\beta}}, \left(X'V^{-1}X + \tau^{-2}I\right)\Big) \tag{15}$$

where $\hat{\boldsymbol{\beta}}$ is denoted in (9).

$$V_i|\cdot \sim \Gamma\left\{\frac{\nu+1}{2},\ \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y - X\beta)^2\right\} \tag{16}$$

$$\sigma^2|\cdot \sim IG\Big(\gamma_1 + \frac{n}{2},\ \gamma_2 + \frac{1}{2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^t(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})\Big) \tag{17}$$

3 Simulation with `tbass()`

We now demonstrate the capabilities of the `TBASS` package using the main command, `tbass()`. For all parameter values of this function, please refer to the help documentation by running `?tbass` after loading the package.

NOTE: At this time, the package `mnormt` is REQUIRED in order to use `TBASS`. Please run `install.packages("mnormt")` to install the package. This dependency will be removed when the package is updated further. No other dependencies are required.

We begin by loading in the package and setting the seed for reproducibility. The package can be installed using the following command: `devtools::install_github("aashen12/TBASS")`.

```
set.seed(12)
library(TBASS)
```

3.1 Friedman Function

Our first example fits the Robust BMARS model to the infamous Friedman Function (Friedman, 1991):

$$10\ sin(\pi x_1 x_2)\ +\ 20(x_3 - 0.5)^2\ +\ 10x_4\ +\ 5x_5 \tag{18}$$

In R:

```
f <- function(x){
  10*sin(pi*x[,1]*x[,2]) + 20*(x[,3]-.5)^2 + 10*x[,4] + 5*x[,5]
}
```

We set our true value of $\sigma = 1$ and attempt to capture the same variability in Robust BMARS

```
sigma <- 1 # TRUE noise sd
n <- 1000 # number of observations
x <- matrix(runif(n*5), nrow = n, ncol = 5)
y <- rnorm(n, mean = f(x), sd = sigma)
```

We then add extra noise to simulate a dataset with outliers, and categorize them for plotting later.

```
ind <- sample(n,size=10) # convert 10 points to outliers
y[ind] <- rnorm(5, f(x[ind,]), 15)

col <- rep(1,n) # for coloring these points in a later plot
col[ind] <- 2
```

From there, we can run the `tbass()` command with 10000 MCMC iterations. Our first simulation is with $\nu = 10$ degrees of freedom to simulate a t-distribution with thicker tails.

```
nmcmc <- 10000 #number of iterations
tb <- tbass(x, y, nu = 10, nmcmc = nmcmc, verbose = FALSE)
```

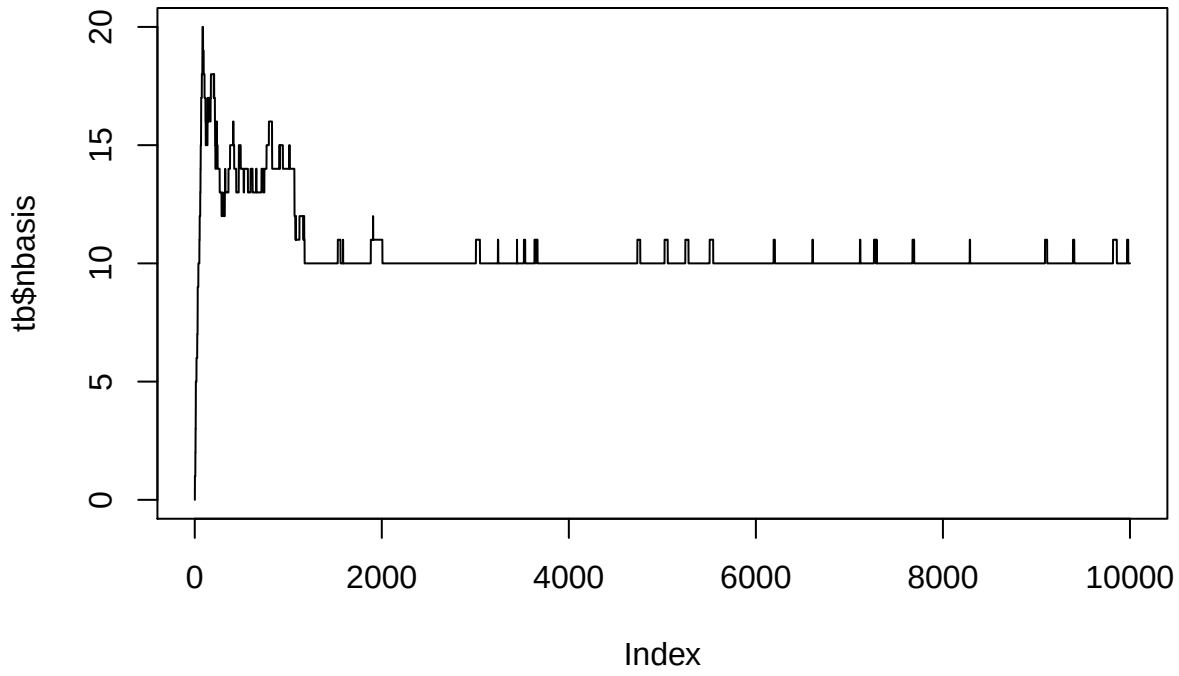
3.2 Results

3.2.1 Overfitting and Prediction

We begin by plotting the number of basis functions throughout the entire simulation.

```
fig <- 3
plot(tb$nbasis, type = "l",
     main = paste0("Figure ", fig, ": Trace plot of number of basis functions"))
```

Figure 3: Trace plot of number of basis functions

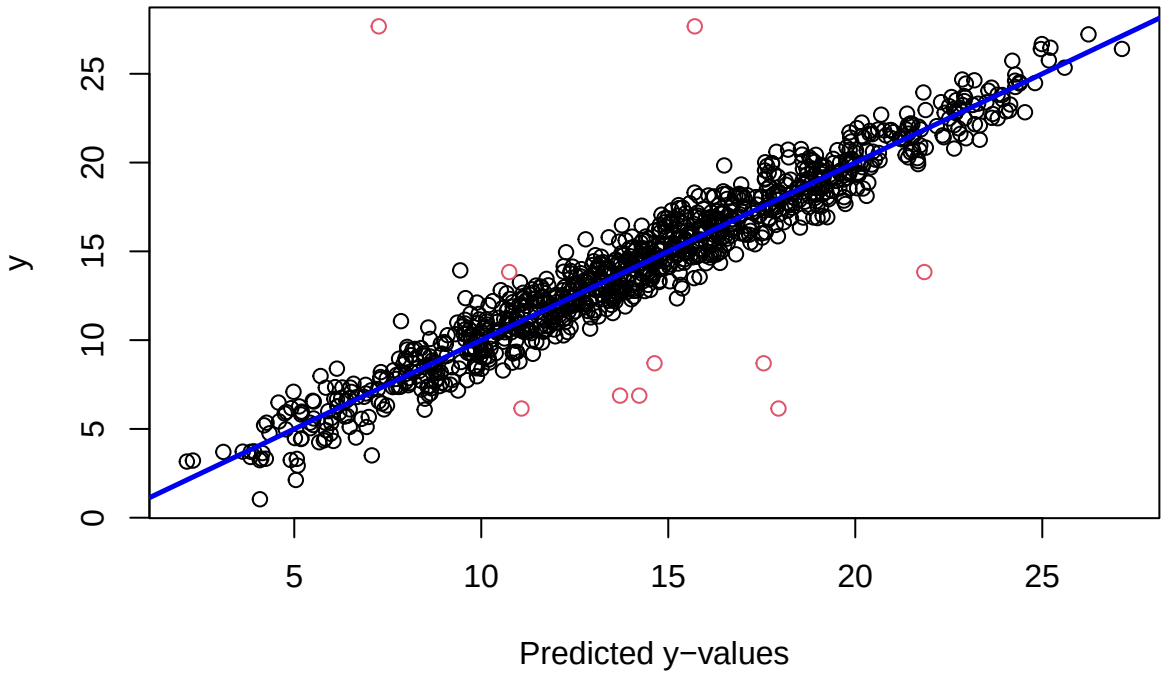


We see that there are 10 basis functions, indicating that there was no overfitting in the `TBASS` model.

We then assess the accuracy of our predicted y values ($X\beta$) vs the actual y-values.

```
fig <- fig + 1
plot(tb$X %*% tb$b, y, col = col,
main=paste0("Figure ", fig, ": Predicted vs Actual Values"),
xlab = "Predicted y-values")
abline(0,1,col="blue2",lwd=2.5)
```

Figure 4: Predicted vs Actual Values



We see that the prediction is quite accurate with respect to the outliers.

3.2.2 Gibbs Results

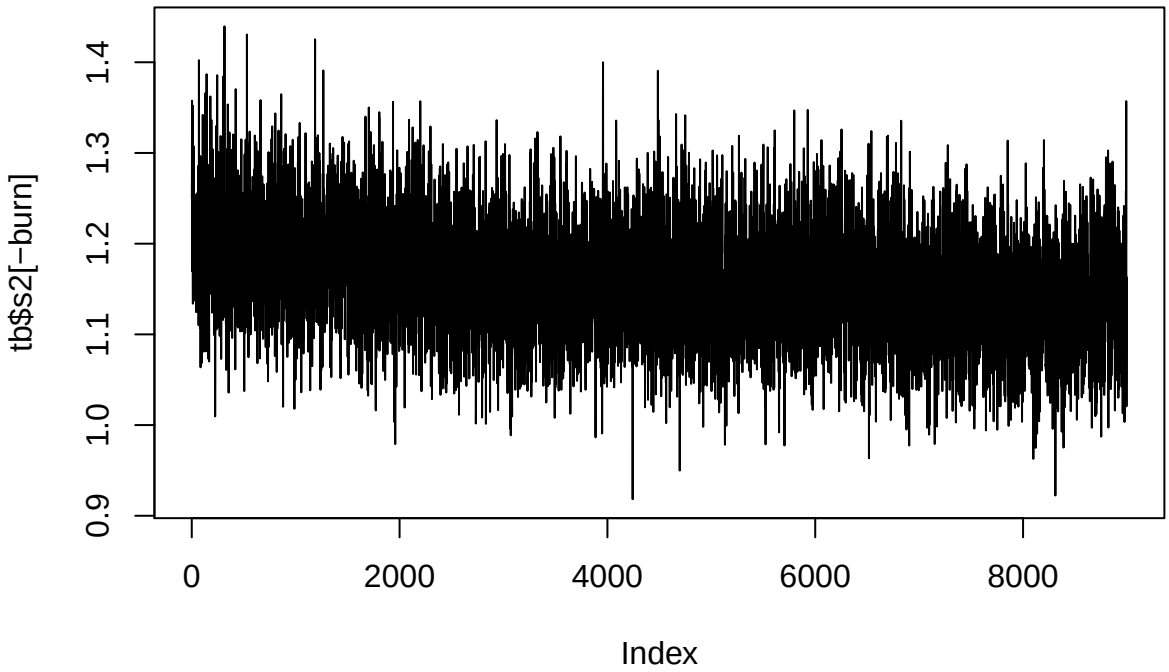
We set a burn-in value to the first 10% of samples.

```
burn_final <- nmcmc/10
burn <- 1:burn_final
```

From there, we plot our σ values to ensure there was no variance inflation.

```
fig <- fig + 1
plot(tb$s2[-burn], type = "l",
    main = paste0("Figure ", fig, ": TBASS Variance Trace Plot"))
```

Figure 5: TBASS Variance Trace Plot

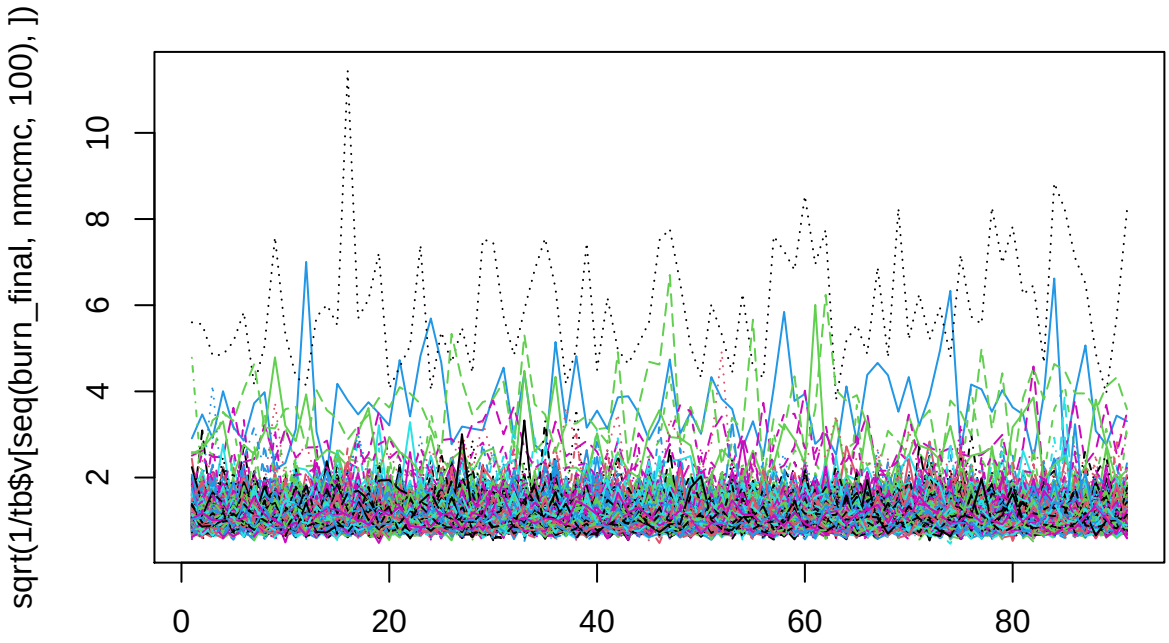


From this plot, we see that our simulated σ^2 values are close to 1, which matches our true value of σ^2 when we generated our data.

We then plot our $\frac{1}{V_i}$ values to see if the outliers are accounted for. We thin every 100 iterations after the burn-in.

```
fig <- fig + 1
matplot(sqrt(1/tb$v[seq(burn_final,nmcmc,100),]), type="l",
  main = paste0("Figure ",fig, ": Plot of V_i"))
```

Figure 6: Plot of V_i



We see that the peaks in V_i are accounting for the outliers, while the majority of V_i values are in the bottom portion of the plot.

3.3 Comparison with BASS

To account for the differences in behavior for Robust BMARS (TBASS) and Gaussian BMARS (BASS), we compare our results from TBASS to the output from BASS, which assumes a Gaussian likelihood.

```
library(BASS)
b <- bass(x,y) # automatically runs 10000 nmcmc iterations
#> MCMC Start #-- Aug 30 14:30:21 --# nbasis: 0
#> MCMC iteration 1000 #-- Aug 30 14:30:24 --# nbasis: 9
#> MCMC iteration 2000 #-- Aug 30 14:30:26 --# nbasis: 9
#> MCMC iteration 3000 #-- Aug 30 14:30:27 --# nbasis: 10
#> MCMC iteration 4000 #-- Aug 30 14:30:29 --# nbasis: 10
#> MCMC iteration 5000 #-- Aug 30 14:30:31 --# nbasis: 10
#> MCMC iteration 6000 #-- Aug 30 14:30:32 --# nbasis: 10
#> MCMC iteration 7000 #-- Aug 30 14:30:34 --# nbasis: 10
#> MCMC iteration 8000 #-- Aug 30 14:30:36 --# nbasis: 10
#> MCMC iteration 9000 #-- Aug 30 14:30:37 --# nbasis: 10
#> MCMC iteration 10000 #-- Aug 30 14:30:39 --# nbasis: 10
```

3.3.1 BASS Variance

```
fig <- fig + 1
plot(b$s2, type = "l",
     main = paste0("Figure ", fig, ": BASS Variance Trace Plot"))
```

Figure 7: BASS Variance Trace Plot