Analysis Plan for Kepler Data

Aliaksandr Hubin

```
#devtools::install_github("https://github.com/jonlachmann/GMJMCMC/tree/FBMS")
library(FBMS)
```

```
## Loading required package: fastglm

## Loading required package: bigmemory

## Loading required package: GenSA

## Loading required package: parallel

## Registered S3 method overwritten by 'FBMS':
##   method     from
##   print.dist stats
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

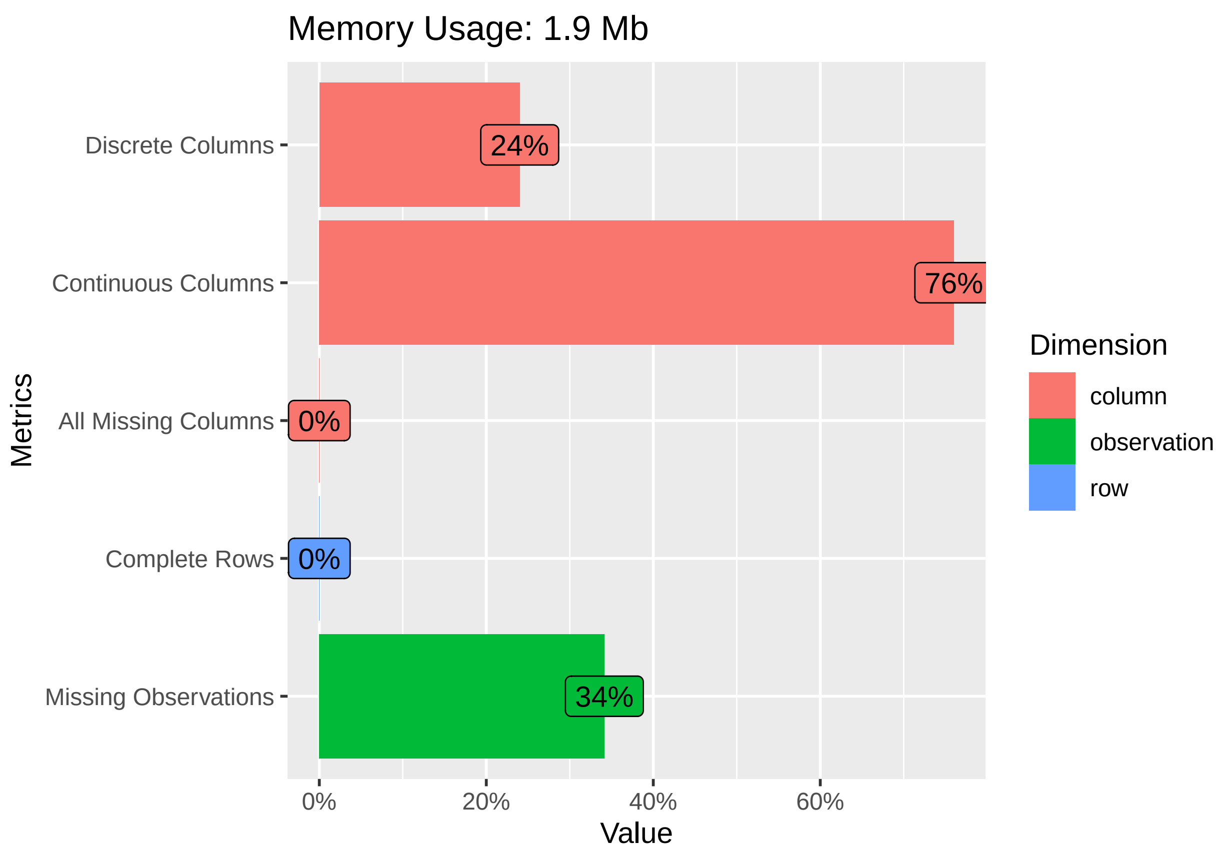
```
library(DataExplorer)
#setwd("Task_Kepler")
```

Load the data and perform EDA

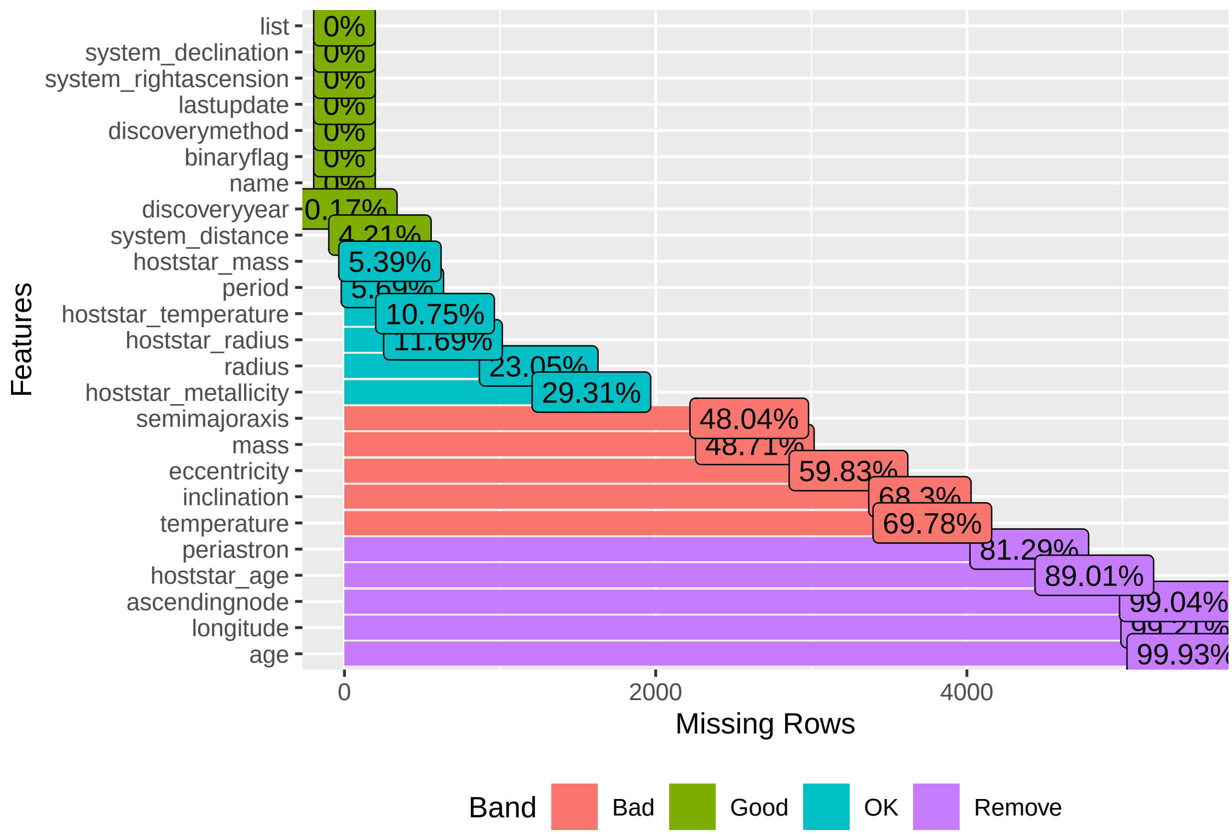
```
data = read.csv("https://raw.githubusercontent.com/OpenExoplanetCatalogue/oec_tables/master/comma_separa

head(data)
```

```
##            name binaryflag mass radius      period semimajoraxis eccentricity
## 1 Kepler-1032 b          0   NA  0.167    3.290118            NA           NA
## 2   HD 154857 b          0 2.24     NA  408.600000         1.291         0.46
## 3   HD 154857 c          0 2.58     NA 3452.000000         5.360         0.06
## 4  Kepler-994 b          0   NA  0.143    1.151167            NA           NA
## 5 Kepler-1350 b          0   NA  0.225    4.496860            NA           NA
## 6 Kepler-1350 c          0   NA  0.154    1.766789            NA           NA
##   periastron longitude ascendingnode inclination temperature age
## 1         NA        NA            NA          NA          NA  NA
## 2         57        NA            NA          NA         336  NA
## 3        352        NA            NA          NA          NA  NA
## 4         NA        NA            NA          NA          NA  NA
## 5         NA        NA            NA          NA          NA  NA
## 6         NA        NA            NA          NA          NA  NA
##   discoverymethod discoveryyear lastupdate system_rightascension
## 1         transit          2016   16/05/10         19 19 43.4040
## 2              RV          2004   14/01/25         17 11 15.7217
## 3              RV          2014   14/01/25         17 11 15.7217
## 4         transit          2016   16/05/10         19 16 17.3254
## 5         transit          2016   16/05/10         19 13 00.1410
## 6         transit          2016   16/05/10         19 13 00.1410
##   system_declination system_distance hoststar_mass hoststar_radius
## 1     +40 05 51.8400         683.854         0.770            0.71
## 2     -56 40 50.8706          64.200         1.718            2.31
## 3     -56 40 50.8706          64.200         1.718            2.31
## 4     +47 24 25.3965         189.186         0.560            0.54
## 5     +46 40 46.5233         343.926         0.550            0.53
## 6     +46 40 46.5233         343.926         0.550            0.53
##   hoststar_metallicity hoststar_temperature hoststar_age              list
## 1                 0.16                 4647           NA Confirmed planets
## 2                -0.31                 5508           NA Confirmed planets
## 3                -0.31                 5508           NA Confirmed planets
## 4                -0.13                 3934           NA Confirmed planets
## 5                -0.06                 3827           NA Confirmed planets
## 6                -0.06                 3827           NA Confirmed planets
```

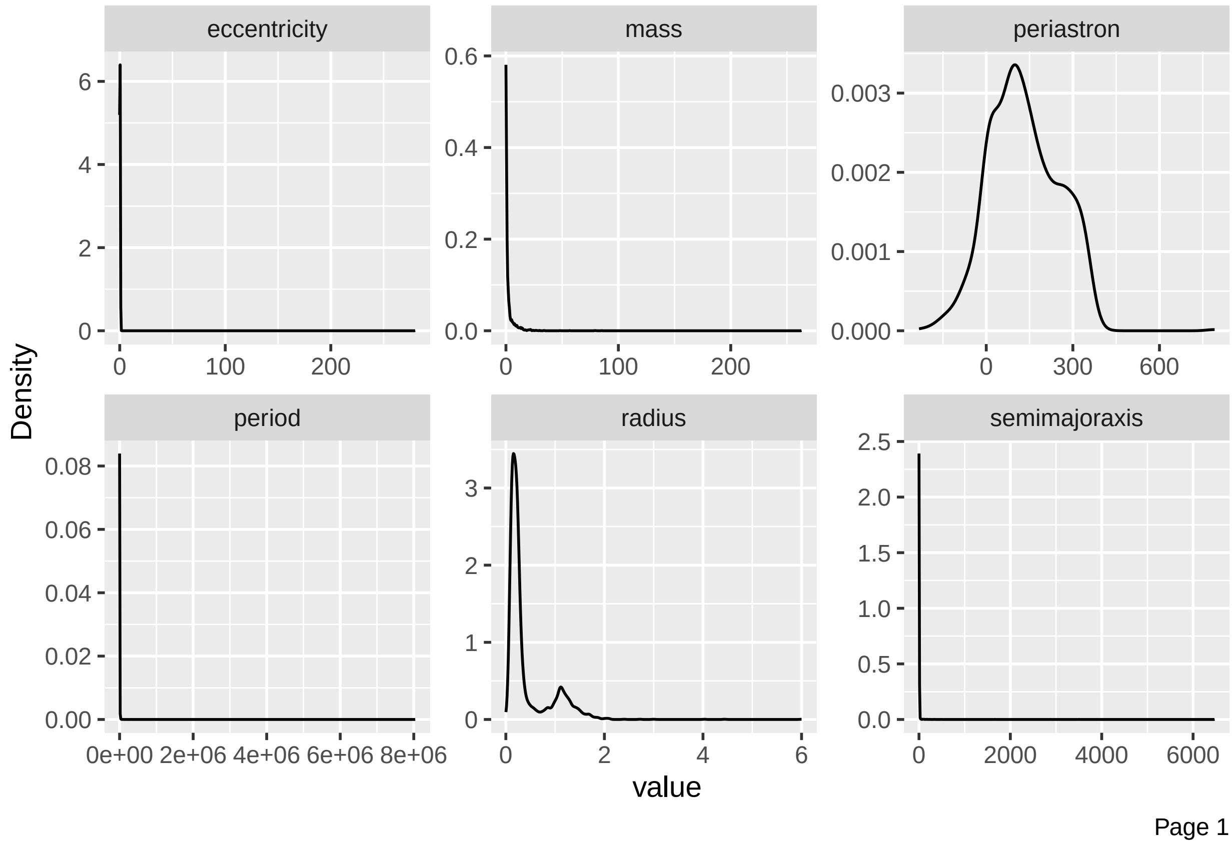
```r
DataExplorer::plot_str(data = data)
DataExplorer::plot_intro(data = data)
```

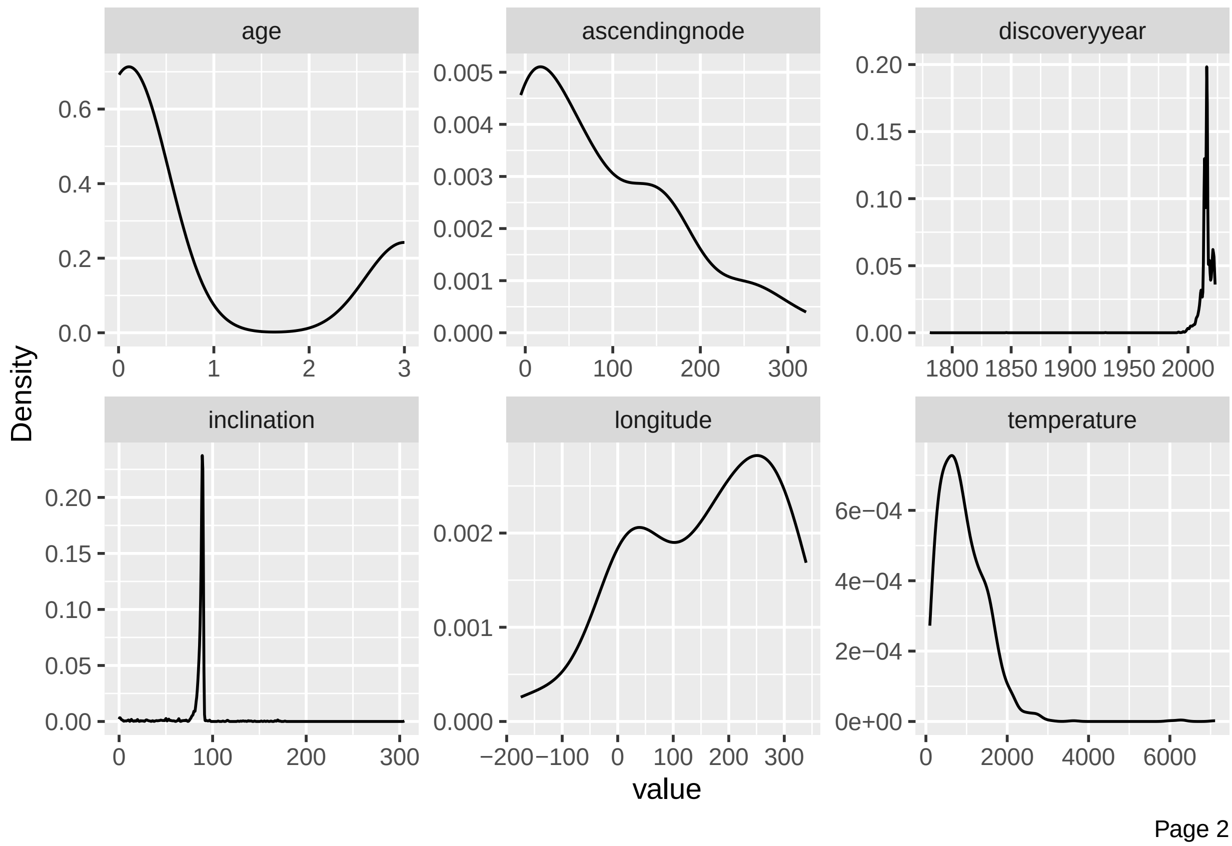


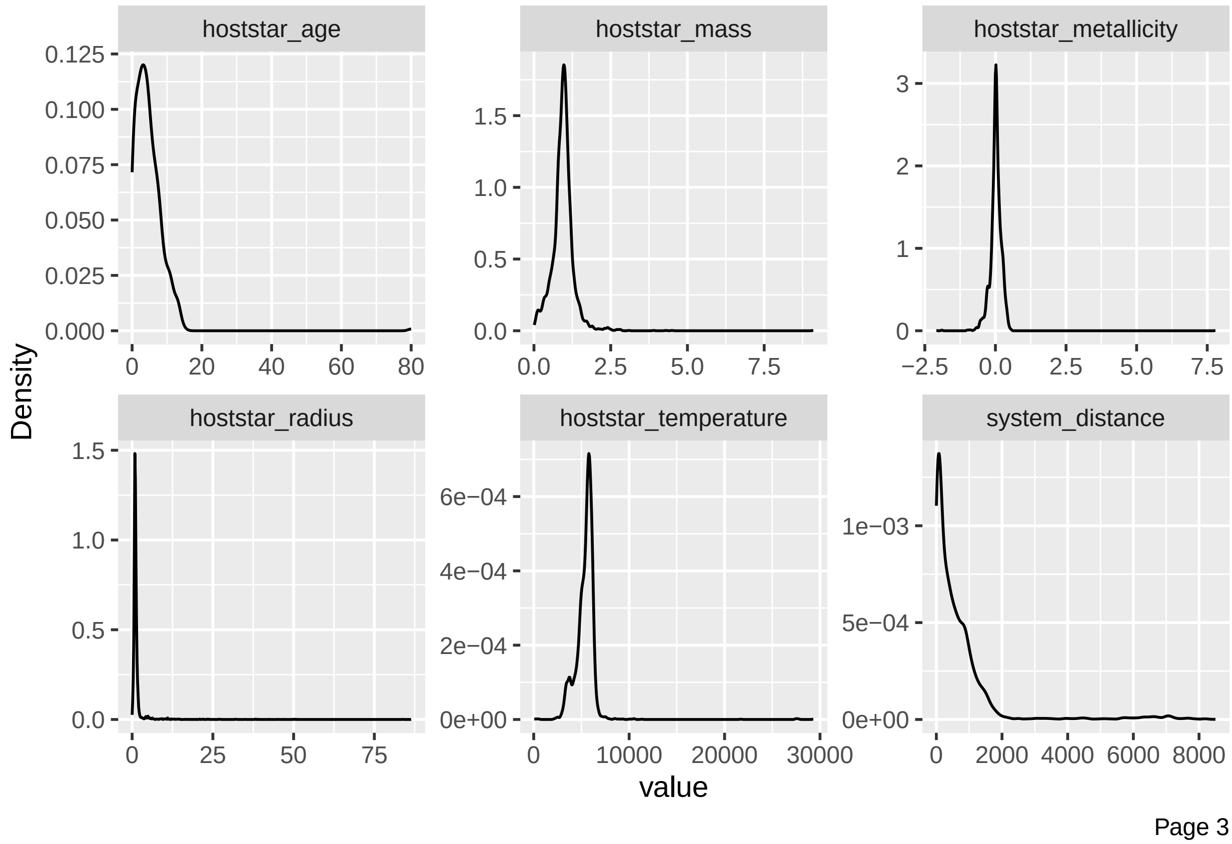

```r
DataExplorer::plot_missing(data = data)
```




```
DataExplorer::plot_density(data[,-2],nrow = 2,ncol = 3)
```

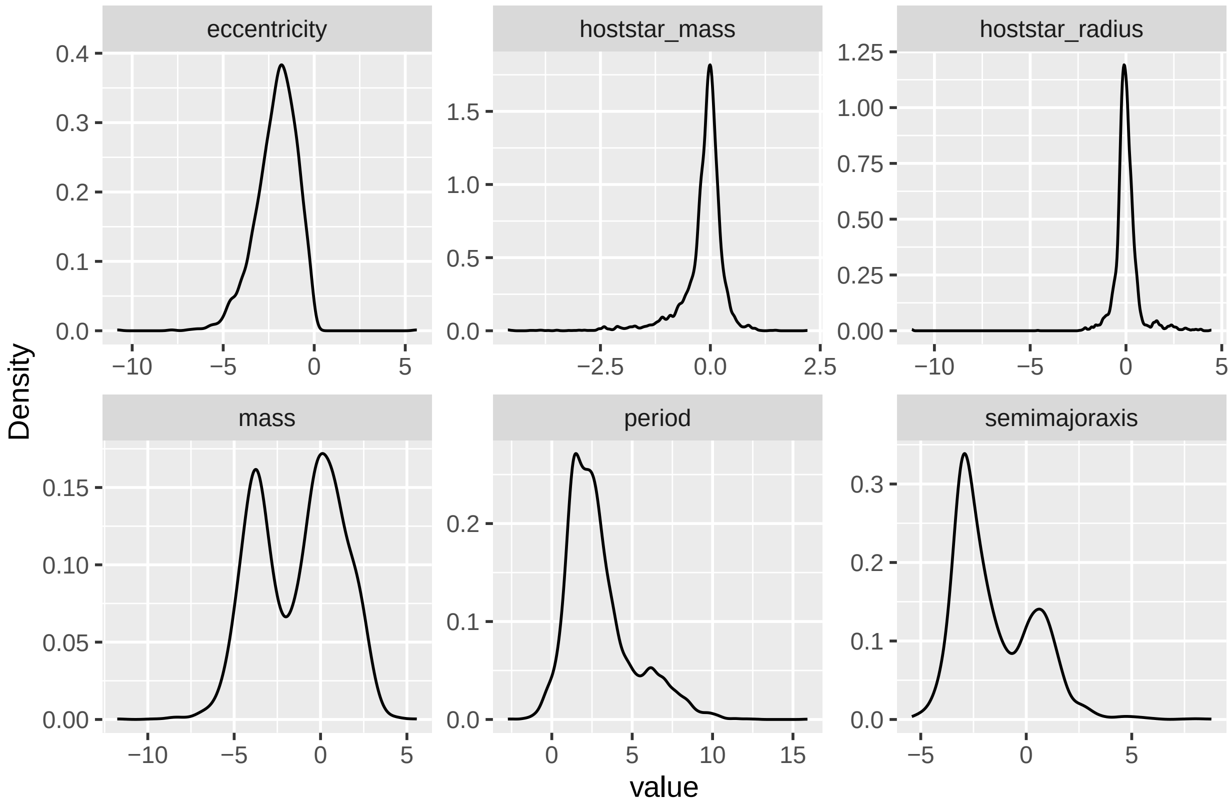

eccentricity
mass
periastron
period
radius
semimajoraxis
Density
value
Page 1


age
ascendingnode
discoveryyear
inclination
longitude
temperature
Density
value
Page 2



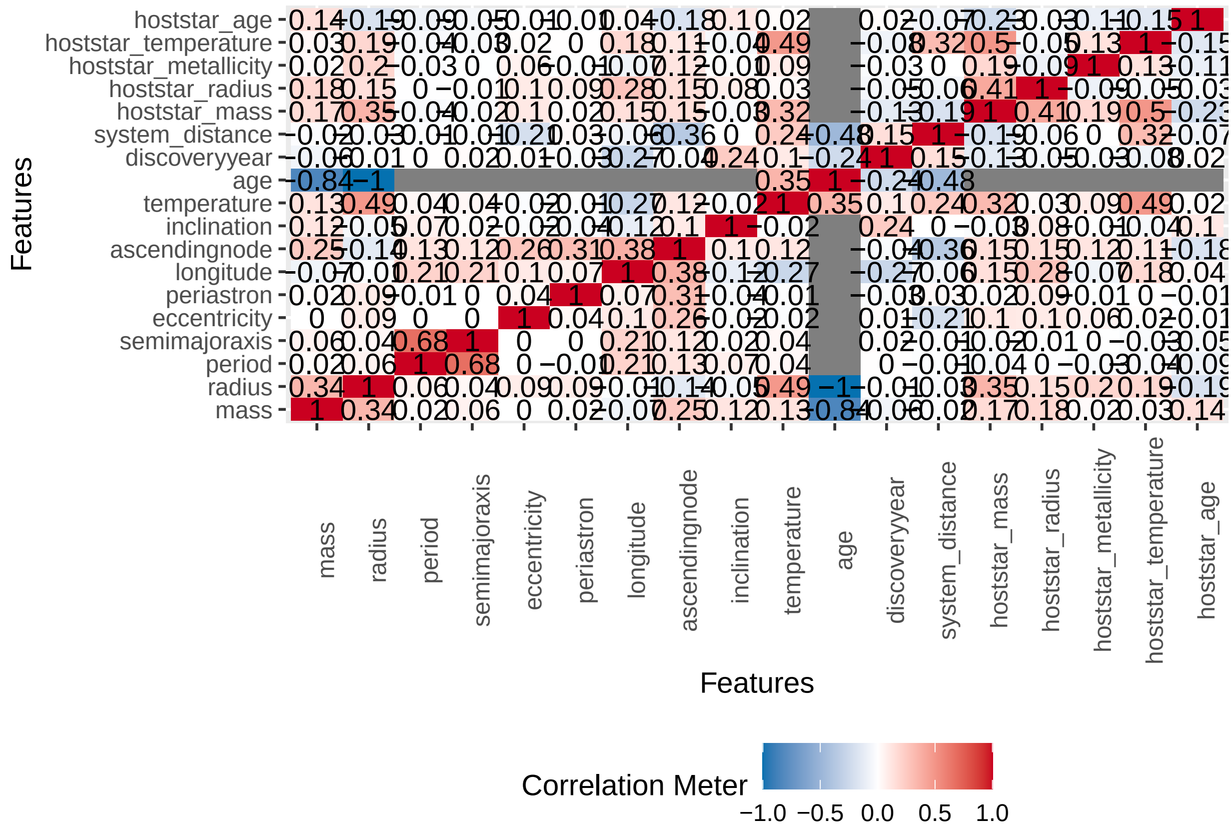

```
DataExplorer::plot_density(log(data[,c("eccentricity","mass","period","semimajoraxis","hoststar_radius"
```

```
## Warning in FUN(X[[i]], ...): NaNs produced
```




```
DataExplorer::plot_correlation(data[,-2],type = "continuous",cor_args = list(use = "pairwise.complete.ol
```

```
## Warning: Removed 24 rows containing missing values (‘geom_text()‘).
```



Select relevant columns for analysis

We shall keep 300 observations as a hold out test set and the rest as the training set

```
data <- data %>% select(semimajoraxis,mass, radius, period, eccentricity, hoststar_mass, hoststar_radius
summary(data)
```

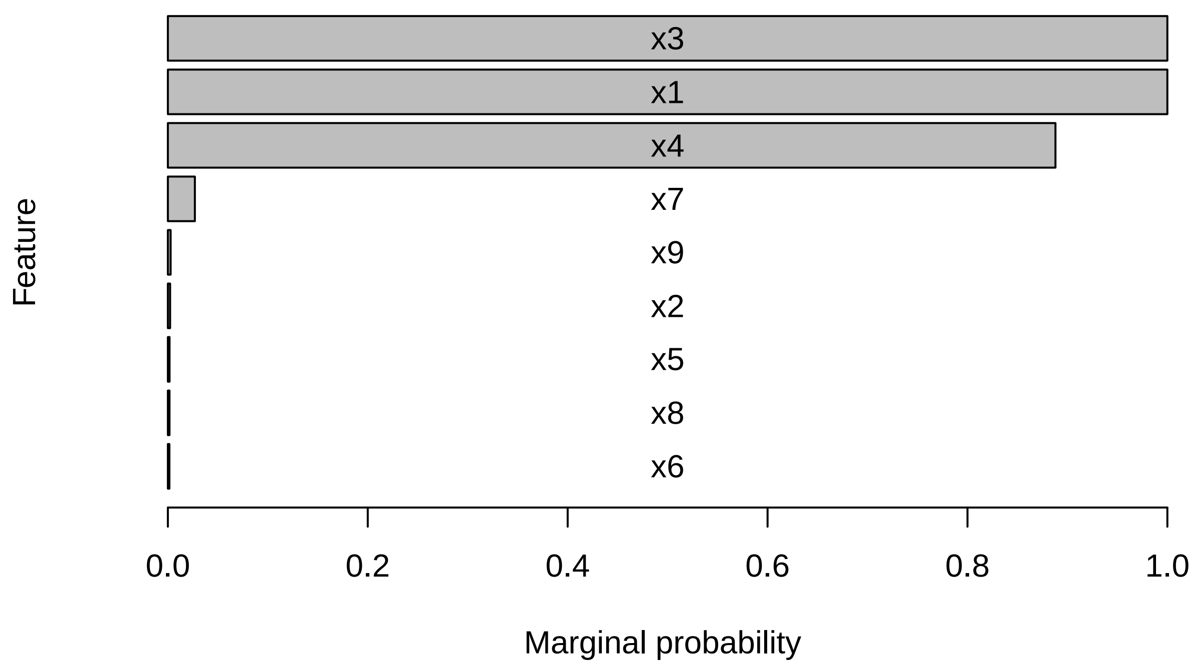
```
##  semimajoraxis          mass              radius            period
##  Min.   : 0.00916   Min.   : 0.00001   Min.   :0.01644   Min.   :     0.45
##  1st Qu.: 0.04328   1st Qu.: 0.02541   1st Qu.:0.22626   1st Qu.:     3.35
##  Median : 0.06010   Median : 0.27300   Median :0.85600   Median :     5.32
##  Mean   : 0.78890   Mean   : 1.27930   Mean   :0.76944   Mean   :  1047.36
##  3rd Qu.: 0.13000   3rd Qu.: 1.05700   3rd Qu.:1.19000   3rd Qu.:    18.36
##  Max.   :67.96000   Max.   :79.00000   Max.   :2.08500   Max.   :166510.00
##   eccentricity       hoststar_mass   hoststar_radius  hoststar_metallicity
##  Min.   :-0.12929   Min.   :0.089   Min.   : 0.121   Min.   :-0.92000
##  1st Qu.: 0.00000   1st Qu.:0.840   1st Qu.: 0.820   1st Qu.:-0.06000
##  Median : 0.00380   Median :1.000   Median : 1.020   Median : 0.06000
##  Mean   : 0.08658   Mean   :1.001   Mean   : 1.309   Mean   : 0.05242
##  3rd Qu.: 0.10938   3rd Qu.:1.173   3rd Qu.: 1.400   3rd Qu.: 0.18100
##  Max.   : 0.93369   Max.   :2.820   Max.   :86.400   Max.   : 7.79000
##  hoststar_temperature   binaryflag
##  Min.   :2516          Min.   :0.00000
##  1st Qu.:5078          1st Qu.:0.00000
```

```
##  Median :5645         Median :0.00000
##  Mean   :5472         Mean   :0.05538
##  3rd Qu.:6000         3rd Qu.:0.00000
##  Max.   :9360         Max.   :2.00000
```

```
set.seed(1)
te.ind <- sample.int(n = 939,size = 300,replace = F)
data.train = data[-te.ind,]
data.test = data[te.ind,]
```

Iteration 1: Simple Bayesian Gaussian regression model with model averaging and Jeffreys prior

```
set.seed(1)
blr <- FBMS::fbms(semimajoraxis ~ ., data = data.train, N = 5000)
plot(blr)
```




```
summary(blr,labels = names(data.train)[-1],effects = c(0.025,0.975),tol = 0)
```

```
##                     Importance | Feature
##  #############################| mass
```

```
##                                | radius
## ##############################| period
##     ##########################| eccentricity
##                               | hoststar_mass
##                               | hoststar_radius
##                               | hoststar_metallicity
##                               | hoststar_temperature
##                               | binaryflag
##
## Best log marginal posterior:  -302.9433
```

```
## $PIP
##          feats.strings   marg.probs
## 1               period 1.000000e+00
## 2                 mass 1.000000e+00
## 3         eccentricity 8.884871e-01
## 4 hoststar_metallicity 2.712699e-02
## 5           binaryflag 2.745115e-03
## 6               radius 1.941690e-03
## 7 hoststar_temperature 1.441157e-03
## 8        hoststar_mass 5.304974e-05
## 9      hoststar_radius 7.253116e-06
##
## $EFF
##               Covariate quant_0.025 quant_0.975
## 1             intercept      0.1169      0.1972
## 2                  mass      0.0679      0.0764
## 3                radius     -0.0804      0.0804
## 4                period       5e-04       5e-04
## 5          eccentricity           0      1.1739
## 6         hoststar_mass           0           0
## 7       hoststar_radius     -0.0804      0.0804
## 8  hoststar_metallicity     -0.4684           0
## 9  hoststar_temperature     -0.0804      0.0804
## 10           binaryflag           0           0
```

Check stability

```r
set.seed(1)
all.probs <- sapply(1:20,FUN = function(x)FBMS::fbms(semimajoraxis ~ ., data = data.train, transforms =

chain_means <- rowMeans(all.probs)
chain_sd <- sapply(1:9, function(i) sd(all.probs[i,]))

alpha_upper <- chain_means + 1.96*chain_sd
alpha_lower <- chain_means - 1.96*chain_sd

stability <- data.frame(covariate = names(data)[-1],one.chain = round(blr$marg.probs[1,],4),mean = roun

print(stability)
```

```
##              covariate one.chain   mean   lower  upper
## 1                 mass    1.0000 1.0000  1.0000 1.0000
```

```
## 2                radius    0.0022 0.0018  0.0006 0.0031
## 3                period    1.0000 1.0000  1.0000 1.0000
## 4          eccentricity    0.8880 0.8888  0.8877 0.8900
## 5         hoststar_mass    0.0017 0.0012 -0.0001 0.0025
## 6       hoststar_radius    0.0015 0.0013  0.0005 0.0021
## 7 hoststar_metallicity    0.0270 0.0263  0.0253 0.0274
## 8 hoststar_temperature    0.0016 0.0012  0.0000 0.0023
## 9            binaryflag    0.0027 0.0023  0.0000 0.0045
```

Marginal Inclusion Probabilities

The marginal inclusion probabilities indicate the likelihood that each predictor is included in the model. High probabilities suggest that the predictor is likely important for explaining the response variable.

1. **period** is included in the model with absolute certainty. The orbital period is a fundamental characteristic of an orbit and strongly influences the semimajor axis due to Kepler's third law, which relates the square of the period to the cube of the semimajor axis.
2. **mass** is also almost certainly included in the model. The mass of the orbiting body can affect the dynamics of the system, influencing the semimajor axis through gravitational interactions.
3. **eccentricity** has a high inclusion probability. The eccentricity of an orbit describes its deviation from a perfect circle, which can affect the semimajor axis as it alters the orbital shape.
4. …
5. **hoststar_mass** has a posterior inclusion below 0.0001 that would indicate it is not important, which is a bit counter-intuitive.

Quantiles of Effect Sizes

The quantiles of model averaged effect sizes of posterior modes across all models provide the 2.5% and 97.5% quantiles for the posterior distribution of each coefficient. We shall look at the predictors with marginal inclusion probabilities above 0.5

1. **mass**: The positive interval indicates that as the mass increases, the semimajor axis is expected to increase. This makes sense physically, as a larger mass could imply a more substantial gravitational influence, potentially affecting the orbit's size.
2. **period**: The very narrow interval indicates a precise positive effect of the orbital period on the semimajor axis, consistent with Kepler's third law.
3. **eccentricity**: The wide interval, spanning from zero to a significant positive value, indicates uncertainty about the effect of eccentricity, but it can have a large positive effect.

   But let us additionally look at the 95% CrI for the median probability model, which under Jeffreys priors approximately correspond to the 95% CI, allowing us to easily obtain the estimates using the lm function in R

   ```r
   summary(lm(semimajoraxis ~ 1 + mass + period + eccentricity, data = data.train))
   ```

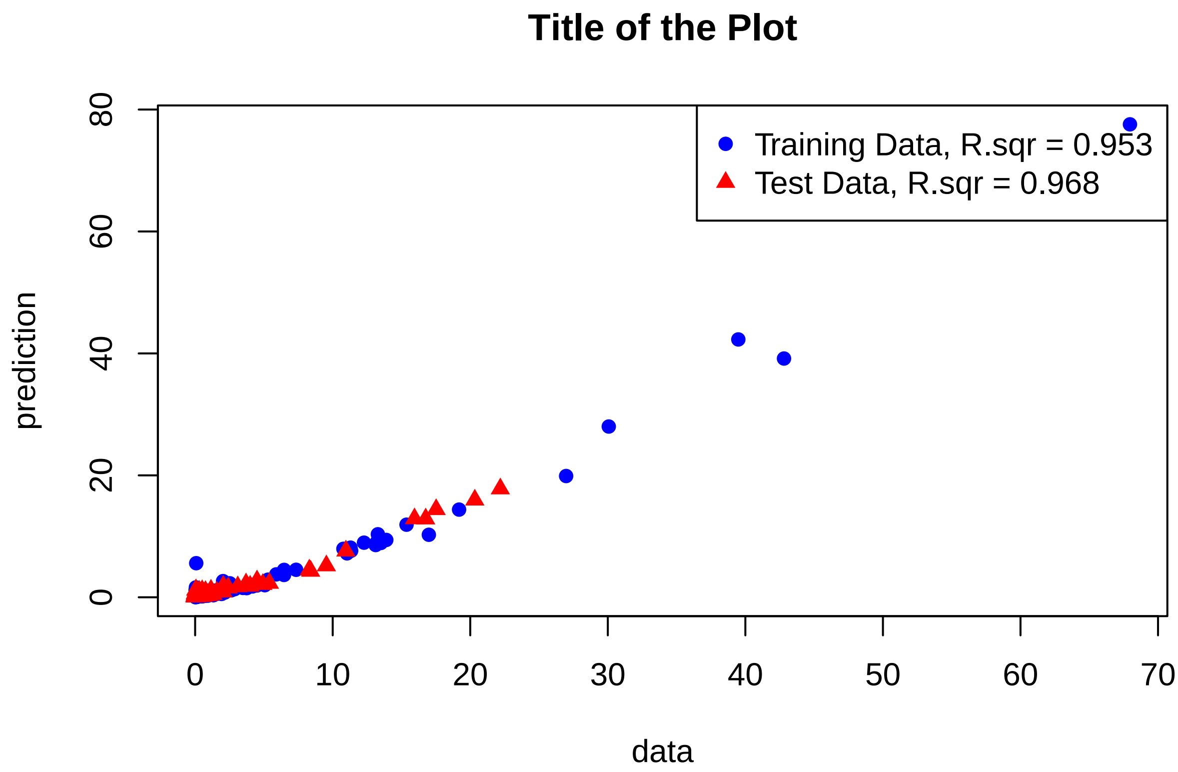
   ```
   ##
   ## Call:
   ## lm(formula = semimajoraxis ~ 1 + mass + period + eccentricity,
   ```

```
##     data = data.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.5954 -0.1971 -0.1097 -0.0410  7.1029
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.169e-01  4.353e-02   2.685  0.00745 **
## mass         6.789e-02  9.293e-03   7.305 8.33e-13 ***
## period       4.630e-04  4.180e-06 110.769  < 2e-16 ***
## eccentricity 1.082e+00  2.479e-01   4.365 1.48e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.945 on 635 degrees of freedom
## Multiple R-squared:  0.953,  Adjusted R-squared:  0.9528
## F-statistic:  4291 on 3 and 635 DF,  p-value: < 2.2e-16
```

essentially supporting the conclusions of model averaged posterior modes with the only exception of that now for the median probability model the 95% CrI does not include 0 for eccentricity. Let us now test predictive ability of the model and then discuss if we are satisfied with it.

Make predictions

```r
preds.train <- predict(blr, data.train[,-1])
preds.test <- predict(blr, data.test[,-1])
r.blr <- round(c(cor(data.train[,1],preds.train$mean)^2,cor(data.test[,1],preds.test$mean)^2),3)
plot(x = data.train[, 1], preds.train$mean, xlab = "data", ylab = "prediction", main = "Title of the Pl
points(x = data.test[, 1], preds.test$mean, col = "red", pch = 17)
legend("topright", legend = c(paste0("Training Data, R.sqr = ",r.blr[1]), paste0("Test Data, R.sqr = ",
```



The predictions show extremely good predictive ability of the model for both training and testing data. Yet, let us reflect on potential pitfalls of the model do decide if we are interested in it.

Possible criticism

The finding that the host star's mass is not important in predicting the semimajor axis does seem counterintuitive, as the mass of the host star should, in theory, have a significant influence on the orbit of the planets around it. Here are some potential reasons and considerations to critically evaluate this finding and whether it motivates the use of a more complex model:

1. **Data Quality and Completeness**:

    - **Missing or Inaccurate Data?** Yet we assumed missing at random for all missing data. But if there are inaccuracies or biases in missing data for host star mass, might it explain why this predictor is not showing importance? We believe it is not important even if we were missing the systems with high or low solar mass systematically as the solar mass would still be much higher than the mass of a planet. The only way it could be important is if the solar mass was constant or having close to zero variance. Yet as we saw in EDA, this is not the case.

2. **Confounding Variables**:

    - **Collinearity**: If host star mass is correlated with other predictors (e.g., period or eccentricity), its unique contribution might be overshadowed, leading to an underestimation of its importance. Again, this is not the case according to EDA.

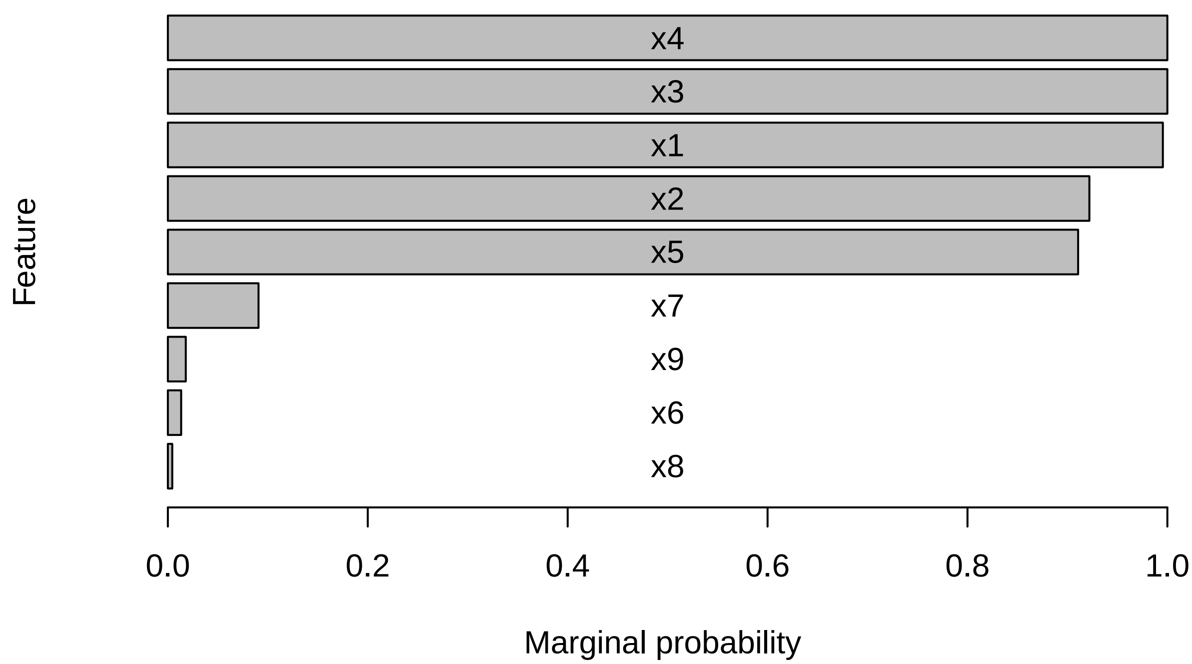
3. **Model Simplicity**:

- **Linear Model Limitations**: A simple linear model might not capture the complex relationships between host star mass and the semimajor axis, particularly if the relationship is non-linear or interacts with other variables. We could try more complex models. Multiplicative effects for the original model could be testes by a model with log transformed responses, while additive non-linearities through fractional polynomials. Finally, more complicated non-linear relationships with both non-linearities and interactions can be tested by a symbolic regression on BGNLM. Let us dig into this.

Iteration 2: Bayesian Gaussian regression model with model averaging and Jeffreys and prior log-transformed response

Inference

```
set.seed(1)
blr.log <- FBMS::fbms(log(semimajoraxis) ~ ., data = data.train, N = 5000)
plot(blr.log)
```




```
summary(blr.log,labels = names(data.train)[-1],effects = c(0.025,0.975))
```

```
##                    Importance | Feature
##  ##############################| mass
##    ############################| radius
## ###############################| period
```

```
## ##############################| eccentricity
##    ###########################| hoststar_mass
##                               | hoststar_radius
##                             ##| hoststar_metallicity
##                               | hoststar_temperature
##                               | binaryflag
##
## Best log marginal posterior:  -416.8585

## $PIP
##          feats.strings  marg.probs
## 1               period 1.000000000
## 2         eccentricity 1.000000000
## 3                 mass 0.995987917
## 4               radius 0.923814291
## 5        hoststar_mass 0.910562576
## 6 hoststar_metallicity 0.090811009
## 7           binaryflag 0.017881853
## 8      hoststar_radius 0.013013003
## 9 hoststar_temperature 0.004386658
##
## $EFF
##               Covariate quant_0.025 quant_0.975
## 1             intercept     -3.0988      -2.694
## 2                  mass      0.0506      0.0496
## 3                radius     -0.8322      0.4048
## 4                period       1e-04       1e-04
## 5          eccentricity      2.2237      2.9948
## 6         hoststar_mass      0.4048      0.3552
## 7       hoststar_radius     -0.4048      0.4048
## 8  hoststar_metallicity      -0.589           0
## 9  hoststar_temperature     -0.4048      0.4048
## 10           binaryflag           0           0
```

1. Inclusion Probabilities:

- **High Inclusion Probabilities:** Period and eccentricity have perfect inclusion probabilities, indicating they are crucial predictors for the log-transformed semimajor axis.
- **Strong Predictors:** Mass, radius, and hoststar_mass also show high inclusion probabilities

2. Effect Sizes:

- **Period:** The very narrow quantiles for period suggest a strong, consistent effect.
- **Eccentricity:** The suggest that eccentricity has a significant positive effect on the log-transformed semimajor axis.
- **Mass and Hoststar Mass:** Both mass and hoststar_mass show positive effects, with hoststar_mass being notably important, which aligns more closely with astrophysical expectations compared to the non-transformed model.
- **Radius:** The interval for radius is wide, indicating more uncertainty in its effect.

- **Other Predictors:** Hoststar_radius, hoststar_metallicity, and hoststar_temperature have wide intervals, suggesting they are less consistent predictors.

  let us also fit the median probability model here

```
summary(lm(log(semimajoraxis) ~ 1 + mass + period + eccentricity + radius + hoststar_mass, data = d
```

```
##
## Call:
## lm(formula = log(semimajoraxis) ~ 1 + mass + period + eccentricity +
##     radius + hoststar_mass, data = data.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.8748 -0.6200 -0.2386  0.3349  4.0209
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -3.061e+00  1.509e-01 -20.282  < 2e-16 ***
## mass           5.040e-02  1.116e-02   4.516 7.51e-06 ***
## period         7.287e-05  4.896e-06  14.882  < 2e-16 ***
## eccentricity   2.590e+00  2.901e-01   8.928  < 2e-16 ***
## radius        -4.499e-01  9.903e-02  -4.544 6.62e-06 ***
## hoststar_mass  7.151e-01  1.694e-01   4.222 2.78e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.102 on 633 degrees of freedom
## Multiple R-squared:  0.4008, Adjusted R-squared:  0.3961
## F-statistic: 84.69 on 5 and 633 DF,  p-value: < 2.2e-16
```

corroborating the model averaged conclusions except for the radius which in MPM has a negative effect, however this makes sense as radius and mass are somewhat correlated and in the models with radius but without mass one would expect a positive effect of the former, hence the uncertainty in the posterior modes.

The log transformation of the response variable appears to improve the model in several ways:

- **Increased Relevance of Predictors:** Variables such as hoststar_mass, which are theoretically important, now have significant inclusion probabilities and effect sizes.
- **Better Fit to Physical Laws:** The transformed model better aligns with astrophysical principles, making the results more interpretable and reliable.

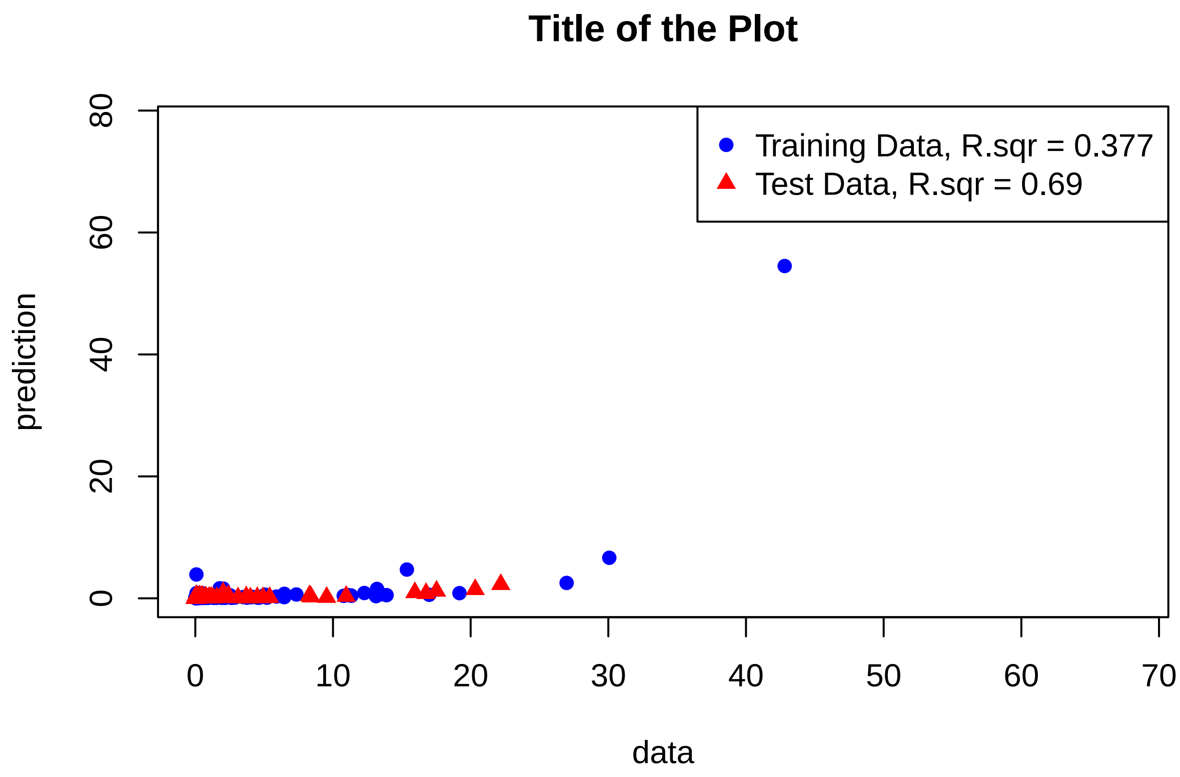
  But let us check the predictive quality of the model.

Predictions

```
preds.train.log <- predict(blr.log, data.train[,-1],link = exp)
preds.test.log <- predict(blr.log, data.test[,-1],link = exp)
r.blr.log <- round(c(cor(data.train[,1],preds.train.log$mean)^2,cor(data.test[,1],preds.test.log$mean)^2
print(r.blr.log)
```

```
## [1] 0.377 0.690
```

```
plot(x = data.train[, 1], preds.train.log$mean,ylim = c(min(preds.train$mean),max(preds.train$mean)), xl
points(x = data.test[, 1], preds.test.log$mean, col = "red", pch = 17)
legend("topright", legend = c(paste0("Training Data, R.sqr = ",r.blr.log[1]), paste0("Test Data, R.sqr =
```

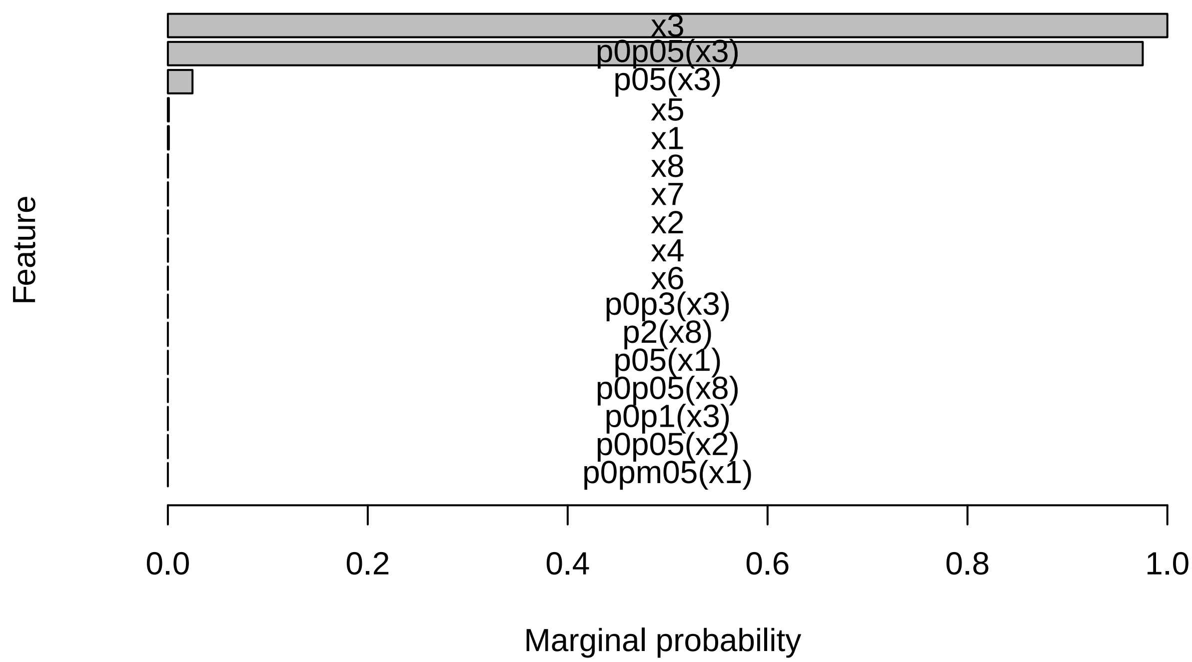


While the log transformation resulted in more interpretatible physically model, it introduced challenges when interpreting predictions on the original scale. The lower predictive quality on the original scale highlights the need for careful consideration of transformation impacts and possibly more sophisticated modeling approaches to improve accuracy. Let us hence dig into Bayesian fractional polynomials.

Iteration 3: More complex functional forms with Bayesian methods

Bayesian fractional polynomials

```
transforms <- c("p0","p2","p3","p05","pm05","pm1","pm2","p0p0","p0p05","p0p1","p0p2","p0p3","p0p05","p0
probs <- gen.probs.gmjmcmc(transforms)
probs$gen <- c(0,1,0,1) # Only modifications!
params <- gen.params.gmjmcmc(data.train)
params$feat$D <- 1   # Set depth of features to 1
set.seed(1)
bfp <- FBMS::fbms(semimajoraxis ~ ., data = data.train,transforms = transforms,runs = 20,cores = 8,P = 
plot(bfp)
```



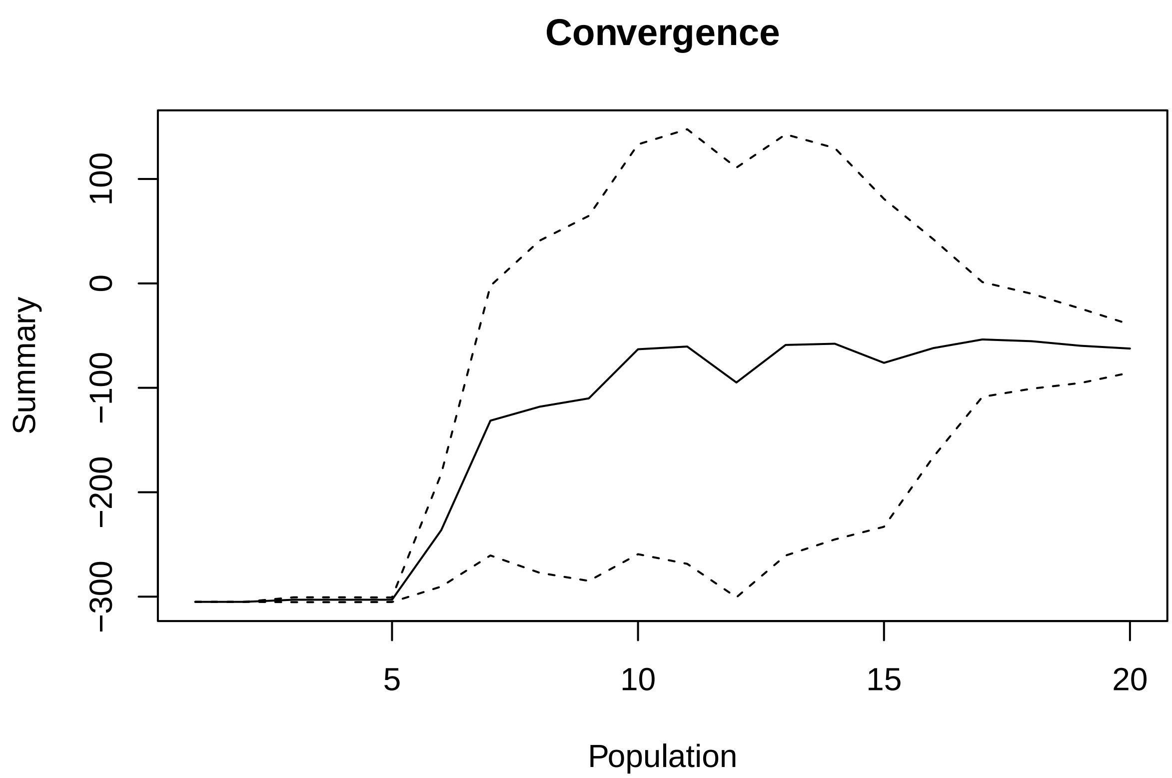

```r
summary(bfp,labels = names(data.train)[-1])
```

```
##                     Importance | Feature
##                                | eccentricity
##                                | radius
##                                | hoststar_metallicity
##                                | hoststar_temperature
##                                | mass
##                                | hoststar_mass
##                                | p05(period)
##   #############################| p0p05(period)
##   #############################| period
##
## Best   population: 19  thread: 1  log marginal posterior: -40.60379
## Report population: 19  thread: 1  log marginal posterior: -40.60379
```

```
##          feats.strings   marg.probs
## 1               period 1.0000000000
## 2        p0p05(period) 0.9753456712
## 3          p05(period) 0.0246543194
## 4        hoststar_mass 0.0009566118
## 5                 mass 0.0009562269
## 6 hoststar_temperature 0.0001903890
## 7 hoststar_metallicity 0.0001600134
## 8               radius 0.0001301342
## 9         eccentricity 0.0001202682
```

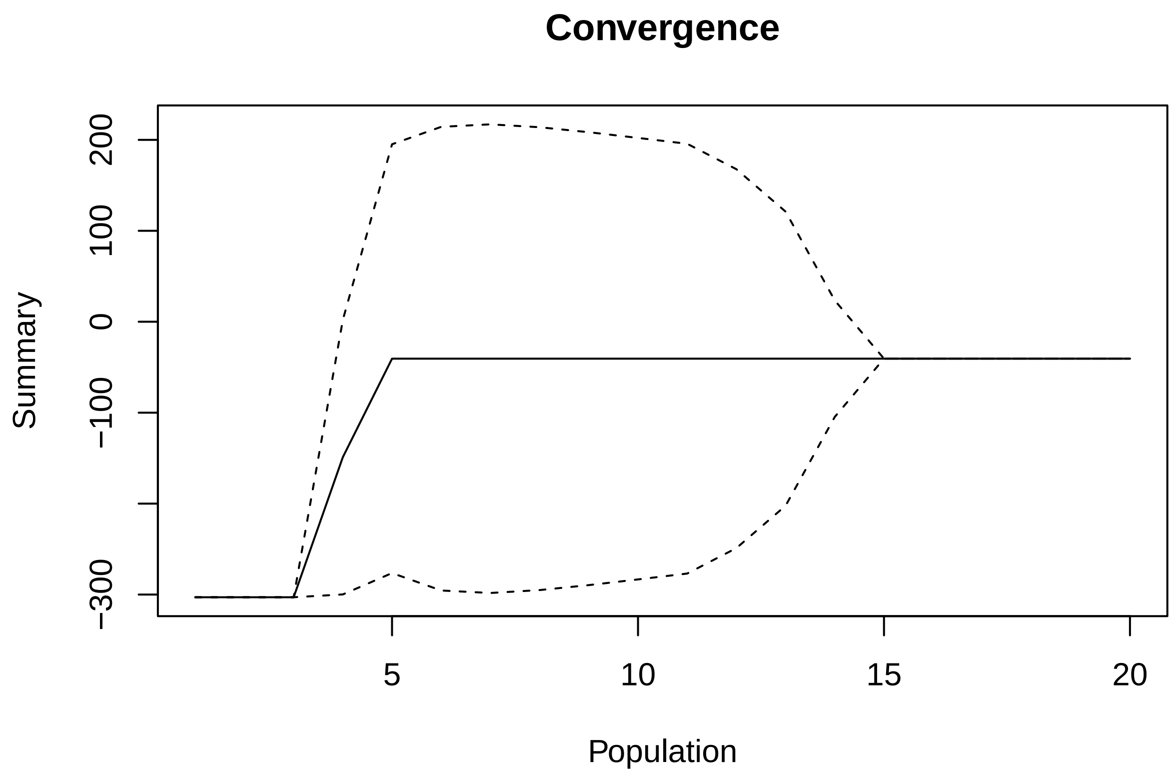
Check convergence

```r
diagn_plot(bfp,window = 10,FUN = median)
```




```
## $stat
##  [1] -304.99231 -304.99231 -302.94327 -302.94327 -302.94327 -236.34918
##  [7] -131.43186 -118.10032 -110.11341  -63.06525  -60.47301  -94.90036
## [13]  -58.94508  -57.78157  -76.07613  -61.99173  -53.69463  -55.35859
## [19]  -59.74723  -62.39649
##
## $lower
##  [1] -304.99231 -304.99231 -305.26194 -305.26194 -305.14295 -290.32626
##  [7] -260.54315 -277.15060 -284.92018 -259.31837 -268.57463 -300.65804
## [13] -260.63587 -245.19847 -233.08357 -166.36847 -108.63835 -100.84193
## [19]  -95.33502  -85.45209
##
## $upper
##  [1] -304.992312 -304.992312 -300.624610 -300.624610 -300.743596 -182.372107
##  [7]   -2.320579   40.949968   64.693353  133.187880  147.628608  110.857332
## [13]  142.745707  129.635318   80.931305   42.385002    1.249084   -9.875252
## [19]  -24.159443  -39.340882
```

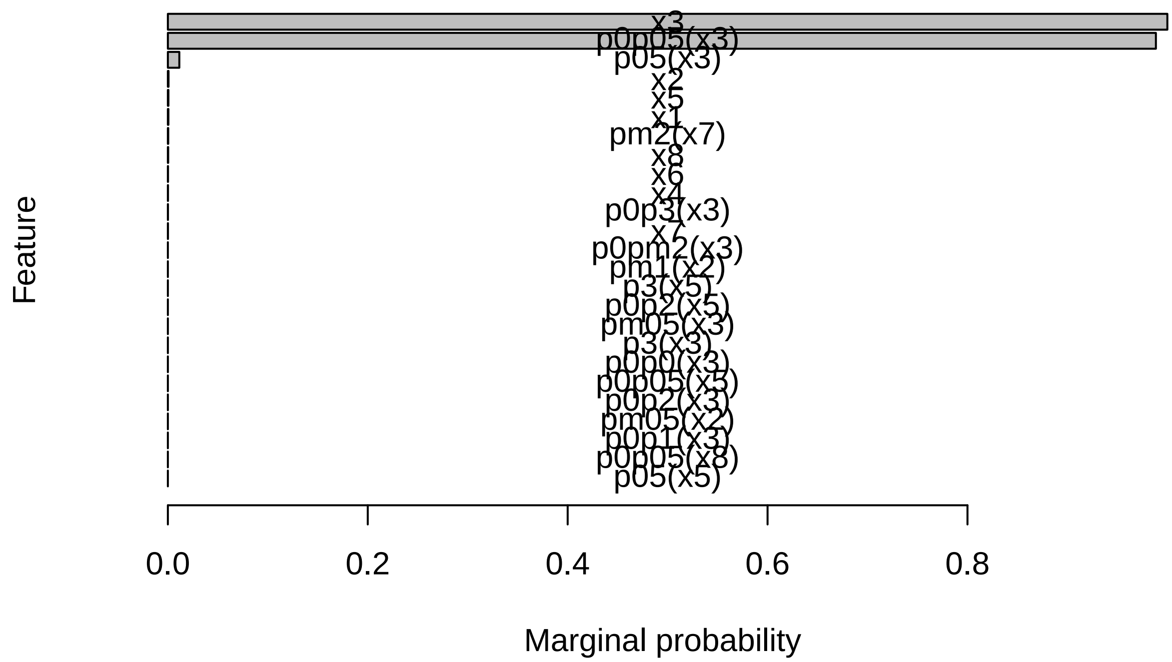
```r
diagn_plot(bfp,window = 10,FUN = max)
```




```
## $stat
##  [1] -302.94327 -302.94327 -302.94327 -148.88022  -40.60379  -40.60379
##  [7]  -40.60379  -40.60379  -40.60379  -40.60379  -40.60379  -40.60379
## [13]  -40.60379  -40.60379  -40.60379  -40.60379  -40.60379  -40.60379
## [19]  -40.60379  -40.60379
##
## $lower
##  [1] -302.94327 -302.94327 -302.94327 -299.85924 -276.36883 -295.54628
##  [7] -298.25053 -295.07118 -289.61215 -283.33223 -276.87609 -248.90163
## [13] -202.29650 -104.58989  -40.60379  -40.60379  -40.60379  -40.60379
## [19]  -40.60379  -40.60379
##
## $upper
##  [1] -302.943273 -302.943273 -302.943273    2.098808  195.161253  214.338703
##  [7]  217.042949  213.863598  208.404566  202.124654  195.668508  167.694056
## [13]  121.088923   23.382313  -40.603790  -40.603790  -40.603790  -40.603790
## [19]  -40.603790  -40.603790
```

We see convergence after 17th generation of GMJMCMC, so let us further increase the number of populations and chains.

```
set.seed(1)
bfp <- FBMS::fbms(semimajoraxis ~ ., data = data.train,transforms = transforms,probs = probs, params = p
plot(bfp)
```




```
summary(bfp,labels = names(data.train)[-1],effects = c(0.025,0.975))
```

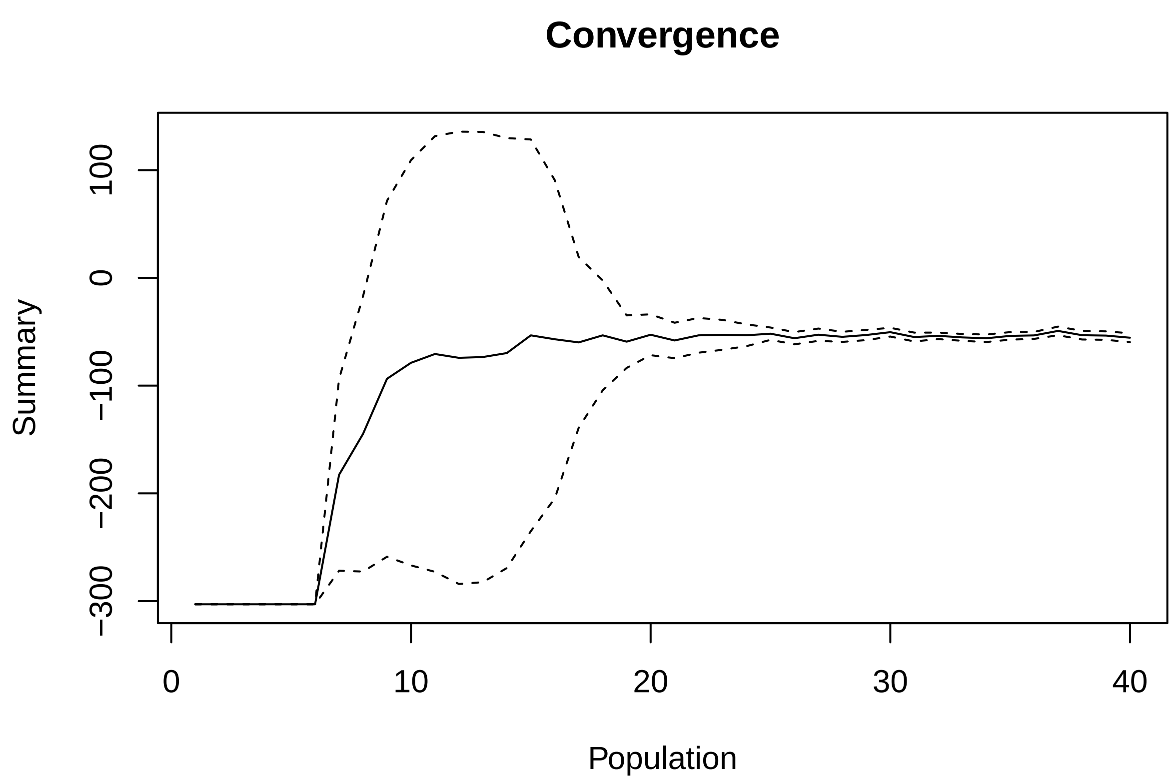
```
##                     Importance | Feature
##                                | hoststar_metallicity
##                                | p0p3(period)
##                                | eccentricity
##                                | hoststar_radius
##                                | hoststar_temperature
##                                | pm2(hoststar_metallicity)
##                                | mass
##                                | hoststar_mass
##                                | radius
##                                | p05(period)
##   #############################| p0p05(period)
##   #############################| period
##
## Best   population: 30  thread: 1  log marginal posterior: -40.60379
## Report population: 30  thread: 1  log marginal posterior: -40.60379
```

```
## $PIP
##                feats.strings   marg.probs
## 1                     period 0.9999999976
## 2              p0p05(period) 0.9885336305
## 3                p05(period) 0.0114665290
## 4                     radius 0.0006661081
```

```
## 5              hoststar_mass 0.0005635466
## 6                       mass 0.0005262631
## 7  pm2(hoststar_metallicity) 0.0003800385
## 8       hoststar_temperature 0.0003245104
## 9            hoststar_radius 0.0001663259
## 10              eccentricity 0.0001440636
## 11              p0p3(period) 0.0001165329
## 12      hoststar_metallicity 0.0001013561
##
## $EFF
##                 Covariate quant_0.025 quant_0.975
## 1               intercept      0.0274      0.0303
## 2                    mass           0           0
## 3                  radius     -0.0029      0.0029
## 4                  period      0.0019       7e-04
## 5            eccentricity     -0.0029      0.0029
## 6           hoststar_mass           0           0
## 7         hoststar_radius     -0.0029      0.0029
## 8  hoststar_metallicity           0           0
## 9  hoststar_temperature     -0.0029      0.0029
## 10             binaryflag           0           0
```

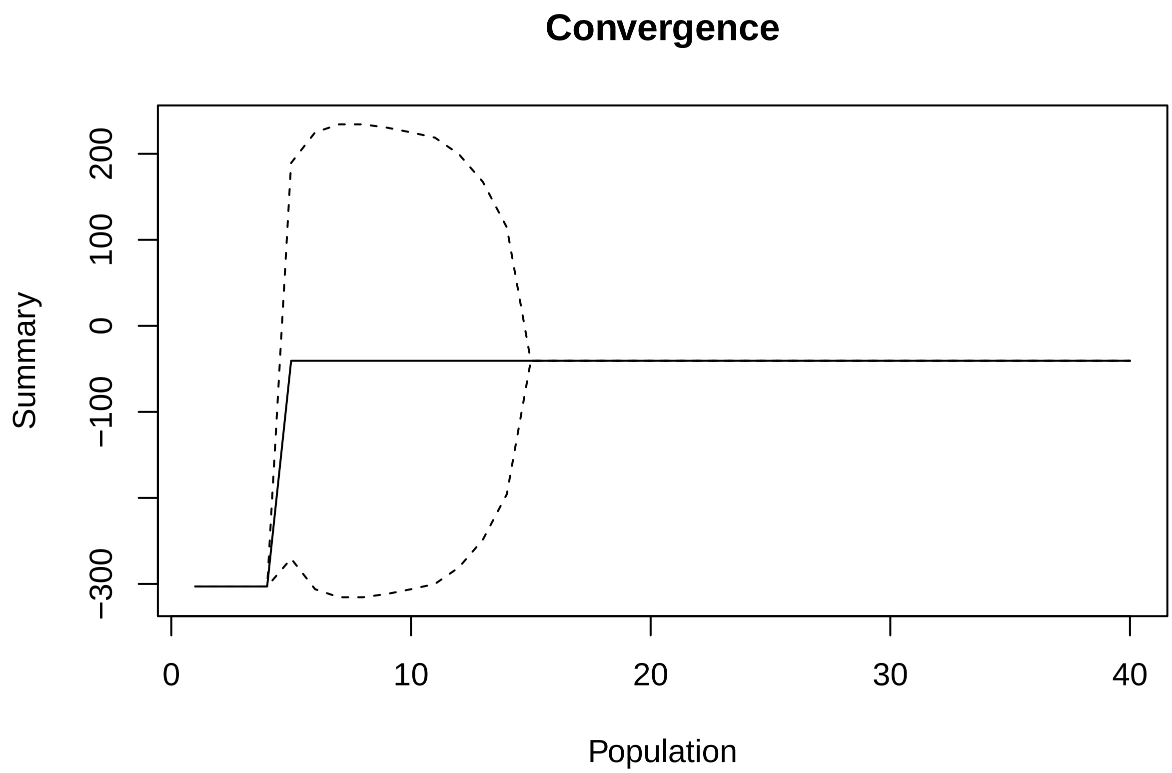
we see the same important effects as in a shorter run. Let us check convergence

```
diagn_plot(bfp,window = 10,FUN = median)
```



```
## $stat
##  [1] -302.94327 -302.94327 -302.94327 -302.94327 -302.94327 -302.94327
##  [7] -182.79259 -144.81009  -93.64555  -78.83091  -70.64733  -74.21637
## [13]  -73.48312  -69.78508  -53.32372  -56.96639  -59.88580  -53.33883
## [19]  -59.19073  -52.80027  -58.08056  -53.32386  -52.85777  -53.28971
## [25]  -51.79900  -56.01227  -52.74895  -54.72501  -53.01538  -50.38987
## [31]  -54.91213  -53.79292  -55.22835  -56.08271  -53.82888  -53.38824
## [37]  -49.23327  -53.20021  -53.54763  -55.53981
##
## $lower
##  [1] -302.94327 -302.94327 -302.94327 -302.94327 -302.94327 -302.94327
##  [7] -271.79983 -272.60998 -258.84264 -266.85056 -272.84206 -284.14994
## [13] -282.44429 -269.32269 -235.15860 -204.64746 -139.04051 -104.35872
## [19]  -83.57190  -71.74518  -74.53105  -69.44804  -66.73479  -63.38263
## [25]  -57.53892  -61.67226  -58.46079  -59.41024  -57.73199  -54.45931
## [31]  -59.02339  -56.79765  -58.43063  -59.58046  -57.30982  -56.62462
## [37]  -53.24591  -57.19375  -57.45488  -59.64880
##
## $upper
##  [1] -302.943273 -302.943273 -302.943273 -302.943273 -302.943273 -302.943273
##  [7]  -93.785356  -17.010187   71.551540  109.188737  131.547398  135.717195
## [13]  135.478052  129.752528  128.511174   90.714684   19.268920   -2.318933
## [19]  -34.809555  -33.855357  -41.630069  -37.199685  -38.980742  -43.196782
## [25]  -46.059072  -50.352287  -47.037101  -50.039775  -48.298778  -46.320431
## [31]  -50.800866  -50.788186  -52.026071  -52.584968  -50.347931  -50.151861
## [37]  -45.220623  -49.206660  -49.640381  -51.430831
```

```r
diagn_plot(bfp,window = 10,FUN = max)
```




```
## $stat
##  [1] -302.94327 -302.94327 -302.94327 -302.94327  -40.60379  -40.60379
##  [7]  -40.60379  -40.60379  -40.60379  -40.60379  -40.60379  -40.60379
## [13]  -40.60379  -40.60379  -40.60379  -40.60379  -40.60379  -40.60379
## [19]  -40.60379  -40.60379  -40.60379  -40.60379  -40.60379  -40.60379
## [25]  -40.60379  -40.60379  -40.60379  -40.60379  -40.60379  -40.60379
## [31]  -40.60379  -40.60379  -40.60379  -40.60379  -40.60379  -40.60379
## [37]  -40.60379  -40.60379  -40.60379  -40.60379
##
## $lower
##  [1] -302.94327 -302.94327 -302.94327 -302.94327 -270.55026 -306.12310
##  [7] -315.44239 -315.44239 -311.59830 -306.12310 -300.01839 -280.77505
## [13] -248.59820 -195.63367  -40.60379  -40.60379  -40.60379  -40.60379
## [19]  -40.60379  -40.60379  -40.60379  -40.60379  -40.60379  -40.60379
## [25]  -40.60379  -40.60379  -40.60379  -40.60379  -40.60379  -40.60379
## [31]  -40.60379  -40.60379  -40.60379  -40.60379  -40.60379  -40.60379
## [37]  -40.60379  -40.60379  -40.60379  -40.60379
##
## $upper
##  [1] -302.94327 -302.94327 -302.94327 -302.94327  189.34268  224.91552
##  [7]  234.23481  234.23481  230.39073  224.91552  218.81082  199.56747
## [13]  167.39062  114.42609  -40.60379  -40.60379  -40.60379  -40.60379
## [19]  -40.60379  -40.60379  -40.60379  -40.60379  -40.60379  -40.60379
## [25]  -40.60379  -40.60379  -40.60379  -40.60379  -40.60379  -40.60379
## [31]  -40.60379  -40.60379  -40.60379  -40.60379  -40.60379  -40.60379
## [37]  -40.60379  -40.60379  -40.60379  -40.60379
```

Convergence seems rather stable for this class of models on this data set and with these tuning parameters of the sampler.

1. Importance of Period:

- The predictor `period` has an extremely high inclusion probability, indicating it is almost certainly a key predictor for the semimajor axis.
- Fractional polynomial transformations of `period`, specifically `p0p05(period)` also show notable inclusion probability. This suggests that non-linear transformations of `period`are important for capturing the relationship with the semimajor axis.

2. Other Predictors: Predictors such as `mass`, `radius`, `hoststar_mass`, `hoststar_metallicity`, `hoststar_temperature`, and `eccentricity` have very low inclusion probabilities, all less than 0.001. This indicates that these variables are not significant predictors in the presence of the `period` and its transformations.

But let us look at MPM here as the model, just like for the model averaged effect, we see positive effect of period and its polynomial term, which makes sense.

```
summary(lm(semimajoraxis ~ 1 + period + p0p05(period), data = data.train))
```

```
##
## Call:
## lm(formula = semimajoraxis ~ 1 + period + p0p05(period), data = data.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2917 -0.0092 -0.0047  0.0010  2.5886
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.157e-02  1.067e-02   2.958  0.00321 **
## period        1.693e-04  3.336e-06  50.763  < 2e-16 ***
## p0p05(period) 7.982e-03  8.384e-05  95.204  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2583 on 636 degrees of freedom
## Multiple R-squared:  0.9965, Adjusted R-squared:  0.9965
## F-statistic: 9.006e+04 on 2 and 636 DF,  p-value: < 2.2e-16
```

Predictions

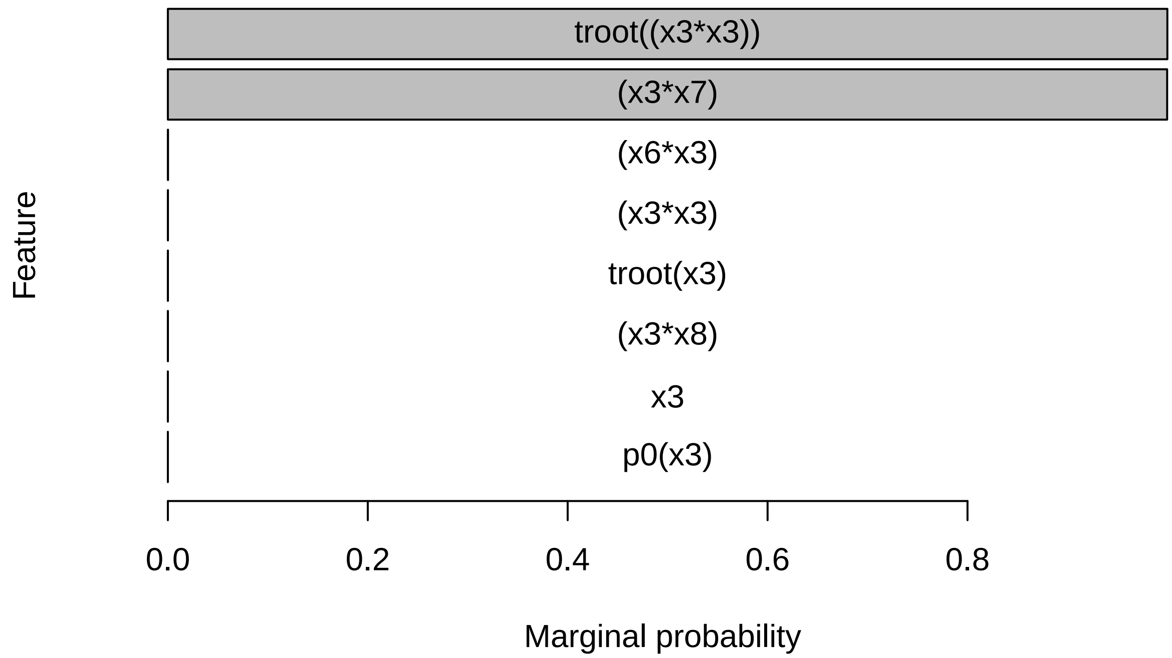
```
preds.train.bfp <- predict(bfp, data.train[,-1])
preds.test.bfp <- predict(bfp, data.test[,-1])
r.bfp <- round(c(cor(data.train[,1],preds.train.bfp$aggr$mean)^2,cor(data.test[,1],preds.test.bfp$aggr$m
plot(x = data.train[, 1], preds.train.bfp$aggr$mean, xlab = "data", ylab = "prediction", main = "Title o
points(x = data.test[, 1], preds.test.bfp$aggr$mean, col = "red", pch = 17)
legend("topright", legend = c(paste0("Training Data, R.sqr = ",r.bfp[1]), paste0("Test Data, R.sqr = ",r
```

The predictions are excellent and even slightly better than for the linear model. Yet, the low inclusion probabilities for predictors such as `hoststar_mass` suggest that these factors do not significantly improve the model once the period is accounted for. This might be due to the fact that the period encapsulates much of the necessary information about the orbit's size and shape. However, it's somewhat surprising that `hoststar_mass` has such a low inclusion probability, as it physically is expected to influence the orbital dynamics. This could indicate that the effect of host star mass is indirectly captured through the period, or it may be due to the specific data set and transformations used.

The limitation of a BFP model is the lack of interactions. So let us try out Bayesian generalized nonlinear models that allow to both model non-linearity and interactions.

Bayesian generalized nonlinear models

```
transforms <- c("sin_deg","exp_dbl","p0","troot","p3")
probs <- gen.probs.gmjmcmc(transforms)
params <- gen.params.gmjmcmc(data.train)
set.seed(1)
bgnlm <- FBMS::fbms(semimajoraxis ~ ., data = data.train,transforms = transforms,runs = 20,cores = 8,P =
plot(bgnlm)
```



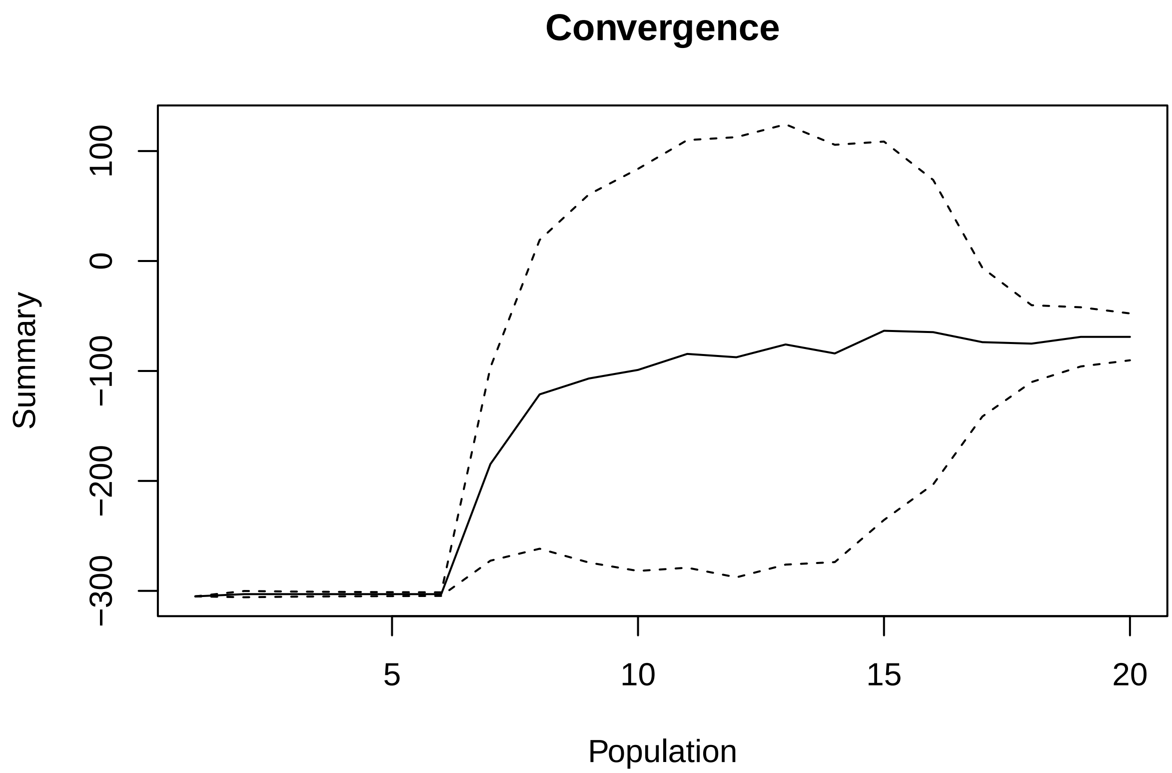

```r
summary(bgnlm,labels = names(data.train)[-1])
```

```
##                     Importance | Feature
##                                | (hoststar_radius*period)
##   #############################| (period*hoststar_metallicity)
##   #############################| troot((period*period))
##
## Best   population: 16  thread: 11  log marginal posterior: -43.41941
## Report population: 16  thread: 11  log marginal posterior: -43.41941
```

```
##                    feats.strings  marg.probs
## 1         troot((period*period)) 0.999954734
## 2 (period*hoststar_metallicity) 0.999737053
## 3      (hoststar_radius*period) 0.000157085
```

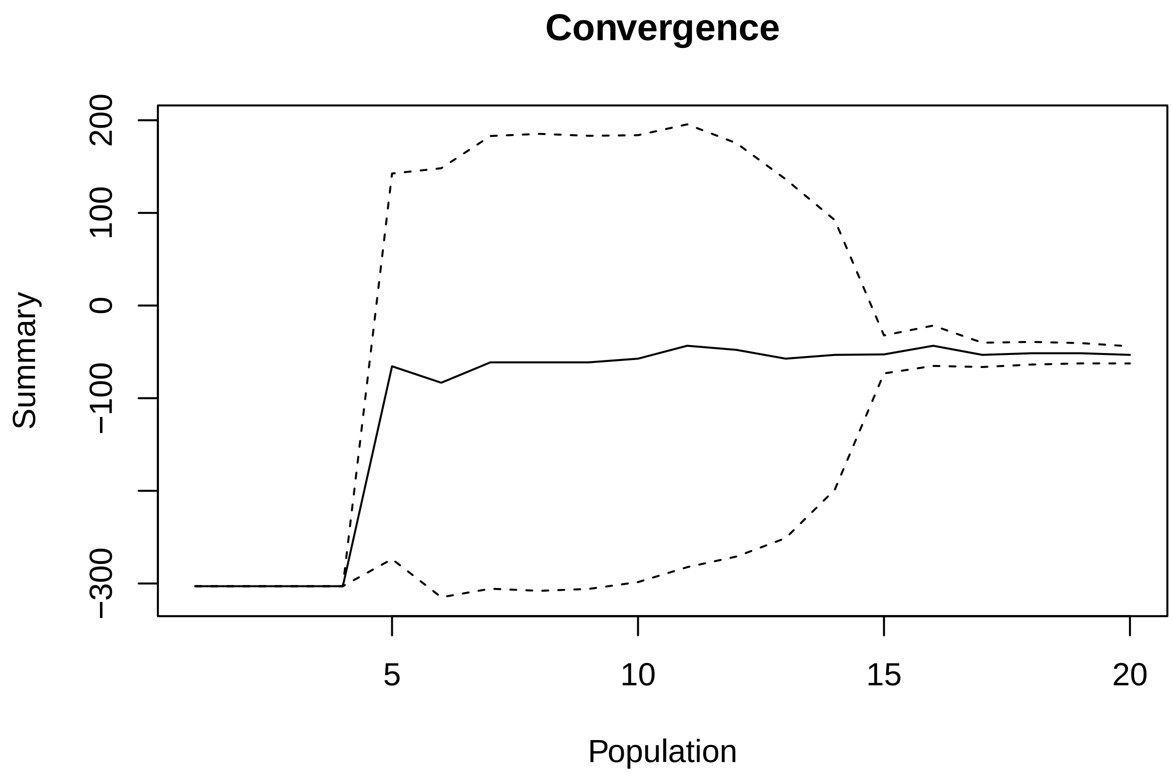
check convergence

```r
diagn_plot(bgnlm,window = 10,FUN = median)
```




```
## $stat
##  [1] -304.99231 -302.94327 -302.94327 -302.94327 -302.94327 -302.94327
##  [7] -184.64588 -121.27432 -106.85044  -99.02986  -84.48624  -87.52671
## [13]  -75.91547  -84.06156  -63.44701  -64.69851  -73.80050  -75.13697
## [19]  -68.98223  -68.98223
##
## $lower
##  [1] -304.99231 -305.78304 -305.26194 -304.95129 -304.73930 -304.58282
##  [7] -272.54593 -261.65040 -274.22129 -281.87282 -278.92555 -287.67863
## [13] -276.11859 -273.81835 -235.53165 -203.12774 -141.40305 -110.08492
## [19]  -95.87906  -90.27668
##
## $upper
##  [1] -304.992312 -300.103502 -300.624610 -300.935252 -301.147244 -301.303731
##  [7]  -96.745821   19.101761   60.520413   83.813095  109.953076  112.625202
## [13]  124.287649  105.695223  108.637636   73.730713   -6.197955  -40.189019
## [19]  -42.085398  -47.687784
```

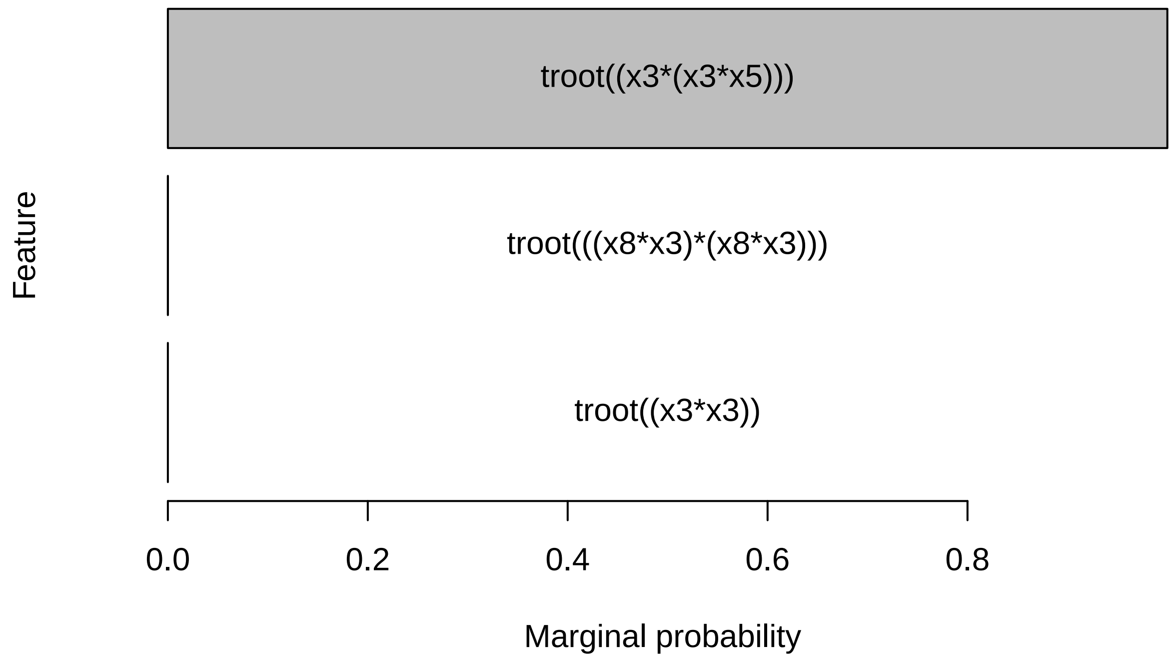
```
diagn_plot(bgnlm,window = 10,FUN = max)
```




```
## $stat
##  [1] -302.94327 -302.94327 -302.94327 -302.94327  -65.57353  -83.32776
##  [7]  -61.31938  -61.31938  -61.31938  -57.32753  -43.41941  -47.82822
## [13]  -57.32753  -53.25107  -52.74216  -43.41941  -53.25107  -51.48473
## [19]  -51.48473  -53.25107
##
## $lower
##  [1] -302.94327 -302.94327 -302.94327 -302.94327 -273.63346 -314.85161
##  [7] -305.64151 -307.94441 -305.84724 -298.50927 -282.40487 -270.94963
## [13] -251.06437 -198.69336  -73.25092  -65.18073  -66.36028  -63.70400
## [19]  -62.43983  -62.49557
##
## $upper
##  [1] -302.94327 -302.94327 -302.94327 -302.94327  142.48640  148.19609
##  [7]  183.00275  185.30566  183.20848  183.85421  195.56604  175.29320
## [13]  136.40930   92.19121  -32.23340  -21.65810  -40.14187  -39.26547
## [19]  -40.52963  -44.00658
```

we see possible convergence for the later generations of GMJMCMC, but let us increase the compute to check if it is indeed so. The limitation of the convergence statistics is that it may show perfect convergence upon algorithm stuck in a good mode and not mixing futher across the modes.

```
set.seed(1)
bgnlm <- FBMS::fbms(semimajoraxis ~ ., data = data.train,transforms = transforms,runs = 64,cores = 8,P
plot(bgnlm)
```

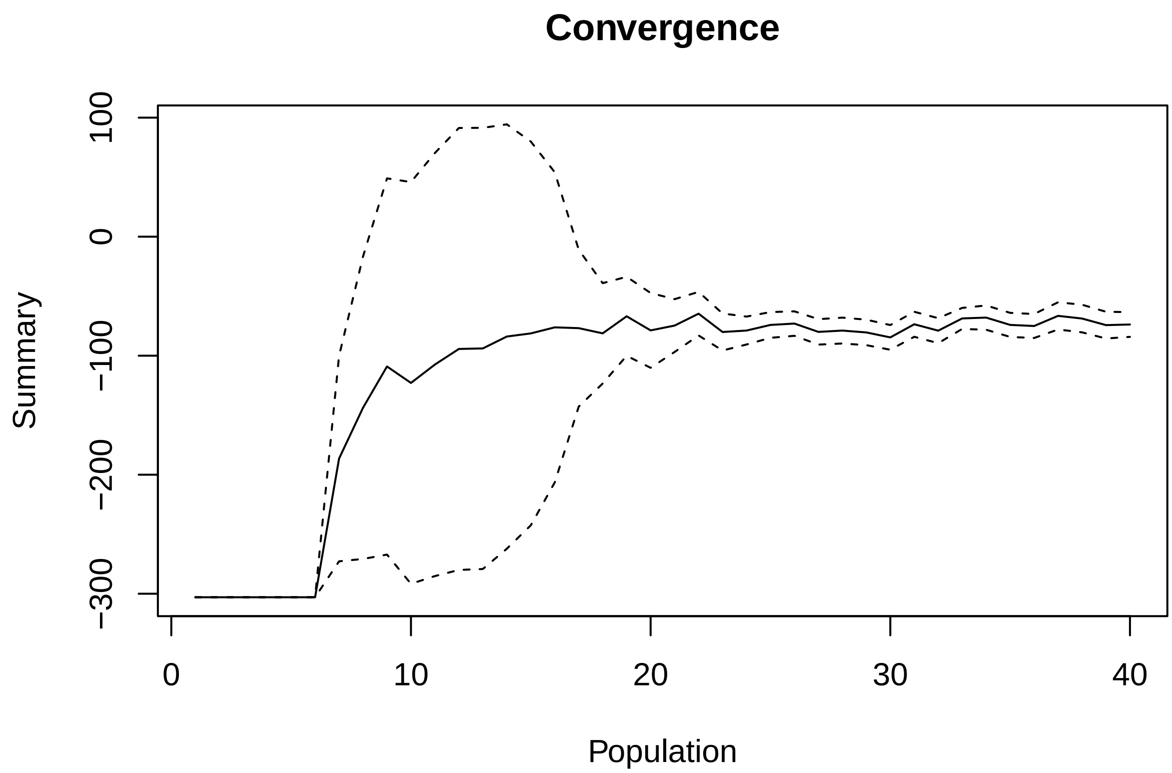



```r
summary(bgnlm,labels = names(data.train)[-1])
```

```
##                      Importance | Feature
##  #############################| troot((period*(period*hoststar_mass)))
##
## Best   population: 14  thread: 4  log marginal posterior: -25.95433
## Report population: 14  thread: 4  log marginal posterior: -25.95433
```

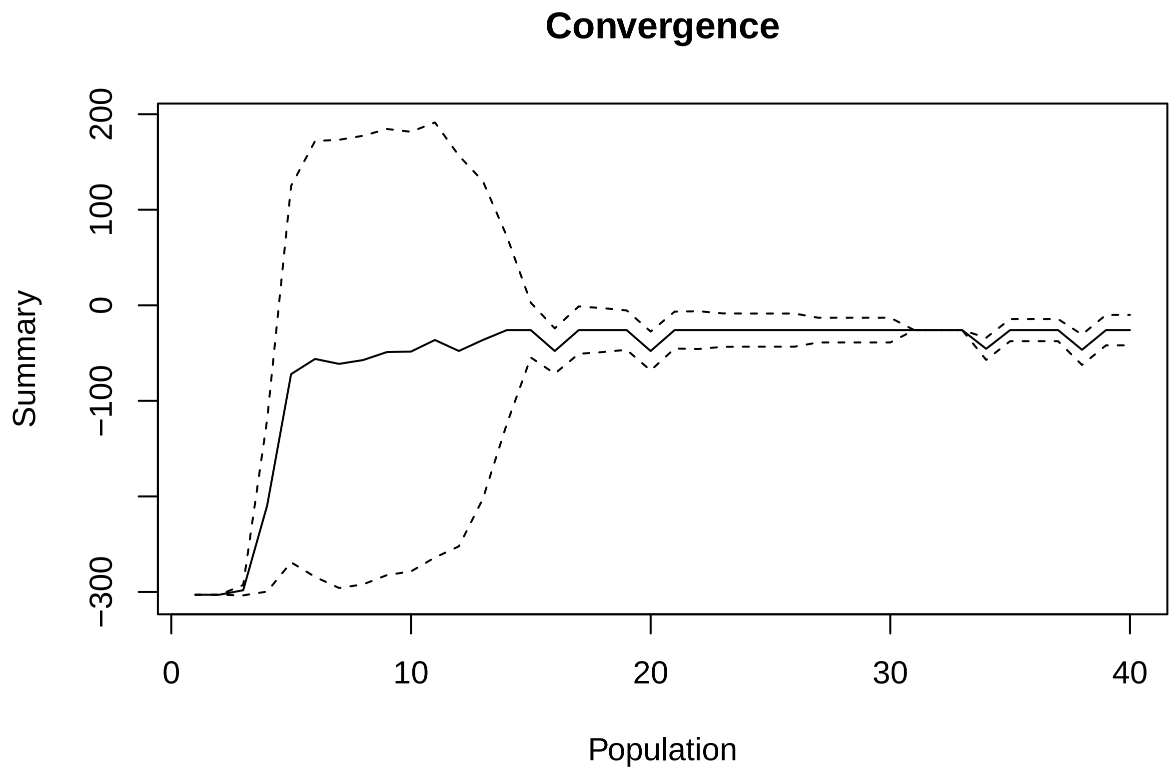
```
##                             feats.strings marg.probs
## 1 troot((period*(period*hoststar_mass)))  0.9999677
```

```r
diagn_plot(bgnlm,window = 10,FUN = median)
```




```
## $stat
##  [1] -302.94327 -302.94327 -302.94327 -302.94327 -302.94327 -302.94327
##  [7] -186.42900 -143.70881 -109.04511 -122.87181 -107.41633  -94.33287
## [13]  -93.88870  -83.89594  -81.27026  -76.17897  -76.86621  -81.21804
## [19]  -66.89314  -78.75126  -74.67980  -64.68262  -80.08704  -78.86655
## [25]  -74.15528  -73.01110  -79.98787  -78.89021  -80.44174  -84.62594
## [31]  -73.59104  -78.97496  -68.67257  -67.98424  -74.11878  -75.04943
## [37]  -66.53810  -68.76453  -74.31500  -73.78164
##
## $lower
##  [1] -302.94327 -302.94327 -302.94327 -302.94327 -302.94327 -302.94327
##  [7] -272.74239 -270.78627 -267.09301 -291.65143 -285.15292 -280.00032
## [13] -279.21836 -262.17043 -242.50790 -206.45858 -142.74790 -123.37913
## [19] -100.08063 -110.16521  -96.85827  -82.98913  -95.63770  -90.60048
## [25]  -84.96068  -83.29605  -90.67397  -89.74986  -91.15488  -94.98894
## [31]  -84.08738  -89.47036  -77.49680  -78.18003  -84.23962  -85.12614
## [37]  -77.97234  -80.37311  -85.54205  -84.18095
##
## $upper
##  [1] -302.94327 -302.94327 -302.94327 -302.94327 -302.94327 -302.94327
##  [7] -100.11560  -16.63135   49.00279   45.90782   70.32025   91.33459
## [13]   91.44096   94.37855   79.96739   54.10064  -10.98453  -39.05696
## [19]  -33.70565  -47.33731  -52.50132  -46.37610  -64.53638  -67.13262
## [25]  -63.34989  -62.72615  -69.30176  -68.03057  -69.72861  -74.26293
## [31]  -63.09471  -68.47956  -59.84834  -57.78845  -63.99793  -64.97272
## [37]  -55.10386  -57.15594  -63.08794  -63.38232
```

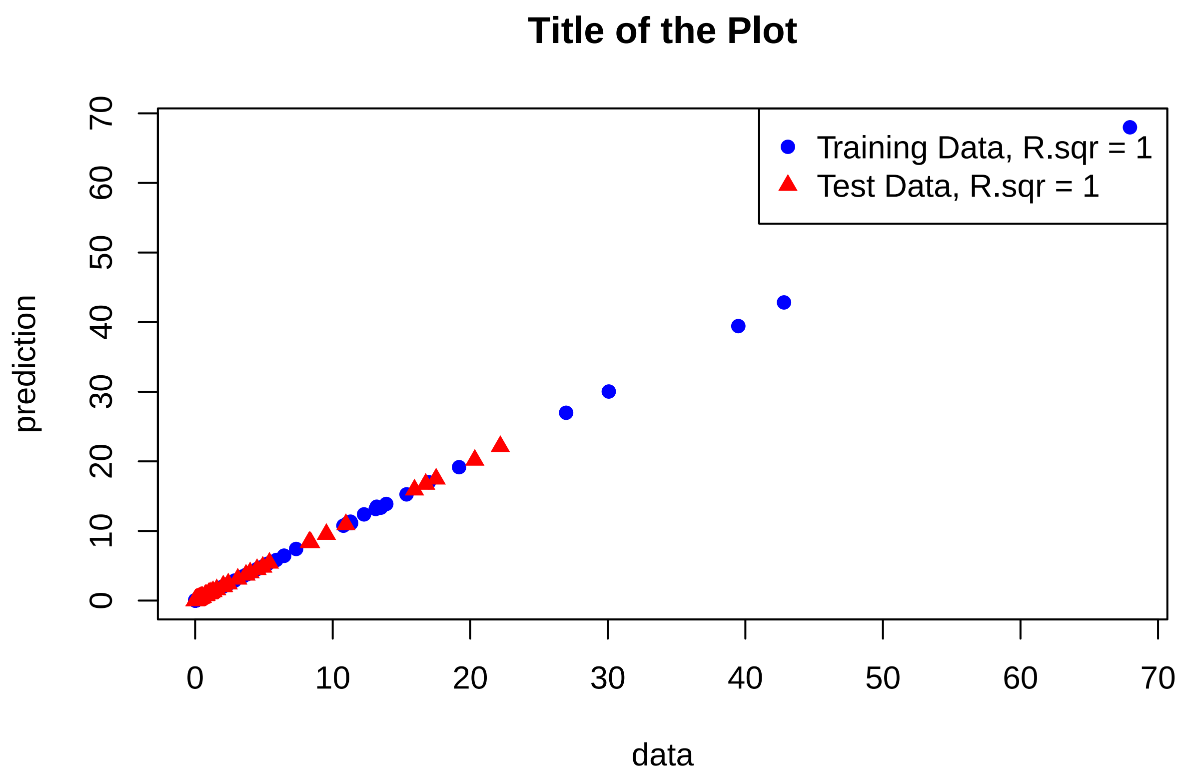
```r
diagn_plot(bgnlm,window = 10,FUN = max)
```




```
## $stat
##  [1] -302.94327 -302.94327 -298.18767 -209.59765  -71.98790  -56.13640
##  [7]  -61.31938  -57.32753  -48.90296  -48.45705  -36.29926  -47.82822
## [13]  -36.29926  -25.95433  -25.95433  -47.82822  -25.95433  -25.95433
## [19]  -25.95433  -47.82822  -25.95433  -25.95433  -25.95433  -25.95433
## [25]  -25.95433  -25.95433  -25.95433  -25.95433  -25.95433  -25.95433
## [31]  -25.95433  -25.95433  -25.95433  -45.54182  -25.95433  -25.95433
## [37]  -25.95433  -46.53073  -25.95433  -25.95433
##
## $lower
##  [1] -302.94327 -302.94327 -303.56904 -299.62849 -269.01243 -284.23486
##  [7] -295.83732 -292.08614 -282.32877 -278.51243 -263.93660 -252.26000
## [13] -202.53770 -124.12450  -54.63168  -71.52444  -50.76811  -48.92999
## [19]  -46.56846  -68.19106  -45.27954  -45.72753  -43.46230  -43.29692
## [25]  -43.29692  -43.29692  -38.88074  -38.88074  -38.88074  -38.88074
## [31]  -25.95433  -25.95433  -25.95433  -57.11707  -37.52959  -37.52959
## [37]  -37.52959  -62.45848  -41.88209  -41.88209
##
## $upper
##  [1] -302.943273 -302.943273 -292.806292 -119.566816  125.036641  171.962072
##  [7]  173.198557  177.431078  184.522850  181.598326  191.338082  156.603564
## [13]  129.939189   72.215832    2.723009  -24.131996   -1.140557   -2.978682
## [19]   -5.340210  -27.465376   -6.629130   -6.181139   -8.446368   -8.611748
```

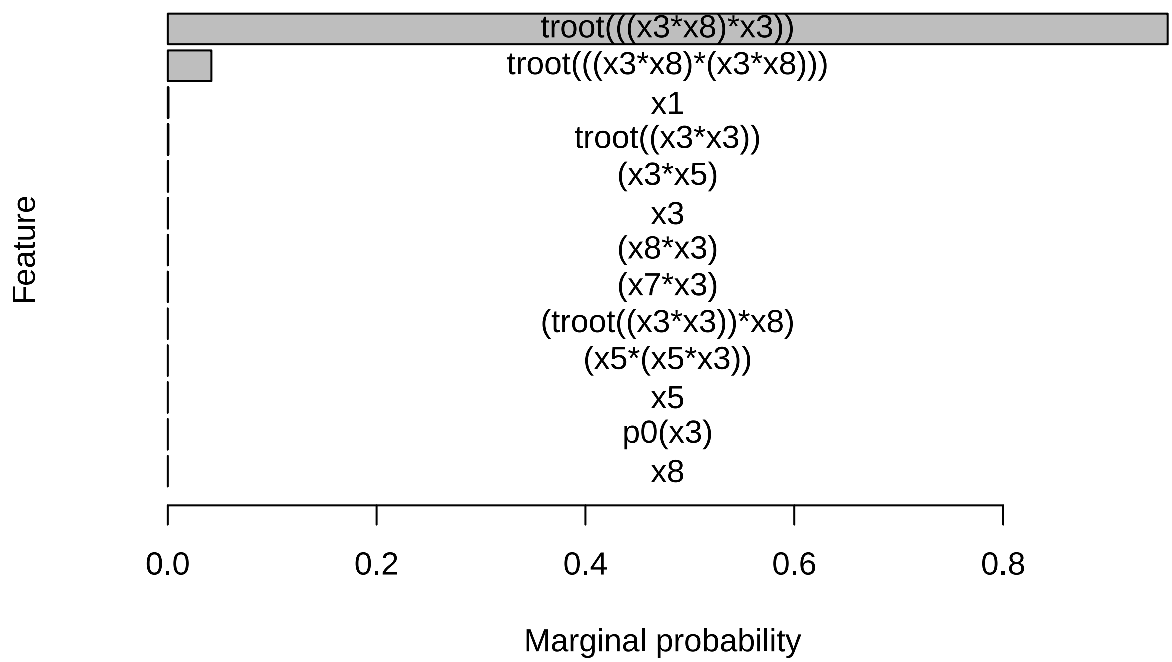
```
## [25]   -8.611748   -8.611748  -13.027934  -13.027934  -13.027934  -13.027934
## [31]  -25.954335  -25.954335  -25.954335  -33.966569  -14.379083  -14.379083
## [37]  -14.379083  -30.602975  -10.026582  -10.026582
```

```
preds.train.bgnlm <- predict(bgnlm, data.train[,-1])
preds.test.bgnlm <- predict(bgnlm, data.test[,-1])
r.bgnlm <- round(c(cor(data.train[,1],preds.train.bgnlm$aggr$mean)^2,cor(data.test[,1],preds.test.bgnlm$
plot(x = data.train[, 1], preds.train.bgnlm$aggr$mean, xlab = "data", ylab = "prediction", main = "Titl
points(x = data.test[, 1], preds.test.bgnlm$aggr$mean, col = "red", pch = 17)
legend("topright", legend = c(paste0("Training Data, R.sqr = ",r.bgnlm[1]), paste0("Test Data, R.sqr =
```



Very good predictions here. But to do a minimal check of reproducibility, let us first rerun the algorithm on the same data

```
set.seed(2)
bgnlm <- FBMS::fbms(semimajoraxis ~ ., data = data.train,transforms = transforms,runs = 64,cores = 8,P =
plot(bgnlm)
```




```r
summary(bgnlm,labels = names(data.train)[-1])
```

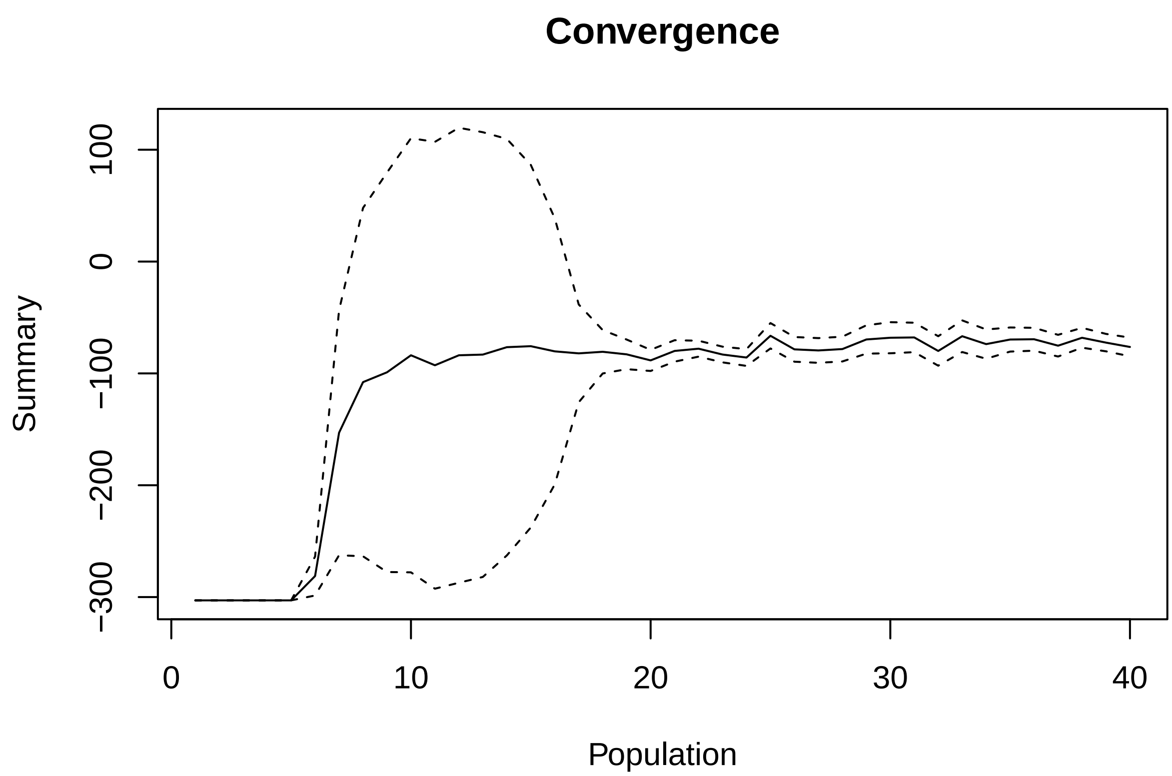
```
##                    Importance | Feature
##                               | (hoststar_temperature*period)
##                               | period
##                               | (period*hoststar_mass)
##                               | troot((period*period))
##                               | mass
##                              #| troot(((period*hoststar_temperature)*(period*hoststar_temperature)))
##   ############################| troot(((period*hoststar_temperature)*period))
##
## Best   population: 24  thread: 58  log marginal posterior: -33.8629
## Report population: 24  thread: 58  log marginal posterior: -33.8629
```

```
##                                                            feats.strings
## 1                          troot(((period*hoststar_temperature)*period))
## 2 troot(((period*hoststar_temperature)*(period*hoststar_temperature)))
## 3                                                                   mass
## 4                                                 troot((period*period))
## 5                                                 (period*hoststar_mass)
## 6                                                                 period
## 7                                           (hoststar_temperature*period)
##     marg.probs
## 1 0.9574076250
## 2 0.0418766252
```

```
## 3 0.0007611380
## 4 0.0007129564
## 5 0.0005592657
## 6 0.0004598099
## 7 0.0001154827
```

Convergence

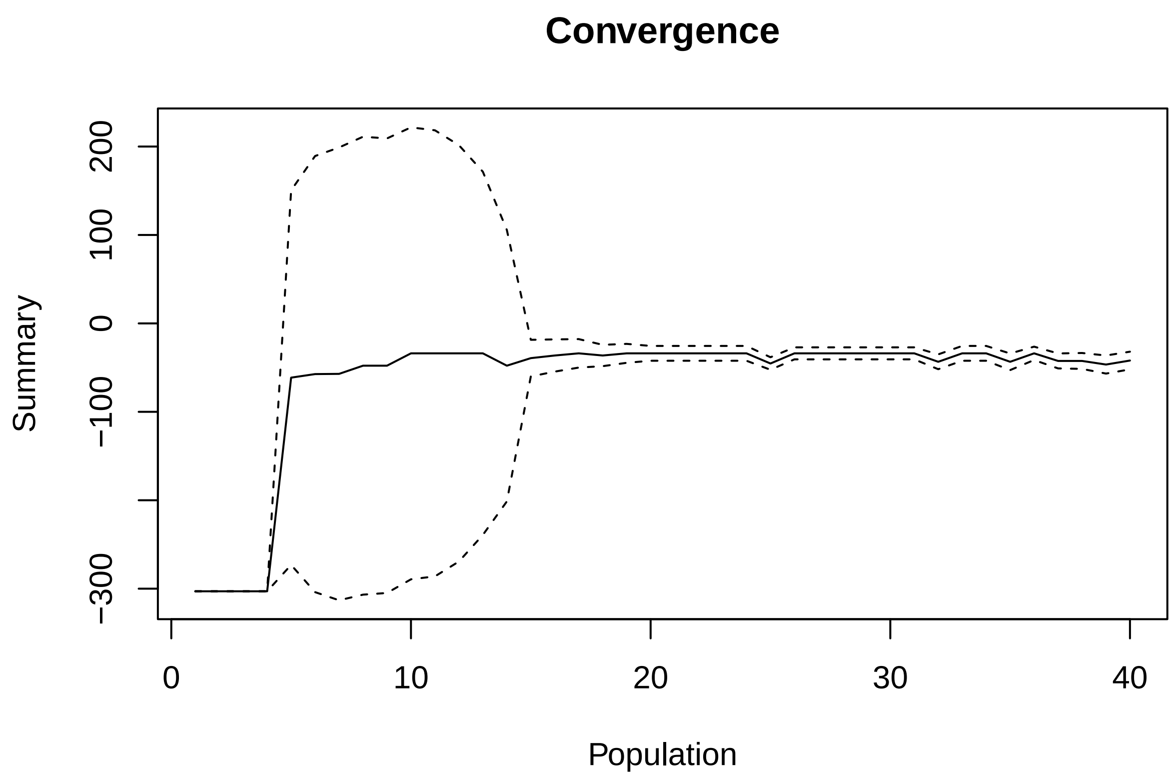
```r
diagn_plot(bgnlm,window = 10,FUN = median)
```




```
## $stat
##  [1] -302.94327 -302.94327 -302.94327 -302.94327 -302.94327 -281.19759
##  [7] -153.02845 -107.79346  -99.01153  -83.78938  -92.72887  -83.78938
## [13]  -83.16634  -76.53822  -75.67336  -80.29303  -82.08097  -80.66582
## [19]  -82.96870  -88.36249  -79.90709  -77.89597  -83.12292  -85.81223
## [25]  -66.34386  -78.51028  -79.49032  -78.24630  -69.66648  -68.11058
## [31]  -67.84644  -79.93131  -66.74696  -73.80941  -69.71001  -69.46443
## [37]  -75.21237  -68.12834  -72.47021  -76.34938
##
## $lower
##  [1] -302.94327 -302.94327 -302.94327 -302.94327 -302.94327 -298.59744
##  [7] -262.55785 -263.55829 -277.60619 -277.87508 -292.51862 -287.21056
## [13] -281.99370 -262.75695 -237.78057 -199.27763 -125.79462 -100.06711
## [19]  -96.20320  -97.82222  -89.47775  -85.00162  -90.13954  -93.30501
## [25]  -77.70313  -89.54630  -90.54348  -89.31694  -82.32357  -82.01635
```

```
## [31]  -81.01448  -93.10219  -80.88355  -86.89564  -80.49164  -79.69605
## [37]  -84.91724  -77.00695  -80.26078  -84.67992
##
## $upper
##  [1] -302.94327 -302.94327 -302.94327 -302.94327 -302.94327 -263.79775
##  [7]  -43.49904   47.97137   79.58314  110.29633  107.06089  119.63180
## [13]  115.66102  109.68052   86.43384   38.69156  -38.36732  -61.26453
## [19]  -69.73420  -78.90276  -70.33644  -70.79032  -76.10629  -78.31946
## [25]  -54.98459  -67.47425  -68.43715  -67.17566  -57.00939  -54.20481
## [31]  -54.67841  -66.76044  -52.61036  -60.72318  -58.92839  -59.23281
## [37]  -65.50750  -59.24974  -64.67964  -68.01885
```

```r
diagn_plot(bgnlm,window = 10,FUN = max)
```




```
## $stat
##  [1] -302.94327 -302.94327 -302.94327 -302.94327  -61.31938  -57.32753
##  [7]  -57.07733  -47.82822  -47.82822  -33.86290  -33.86290  -33.86290
## [13]  -33.86290  -47.82822  -39.27938  -36.29926  -33.86290  -36.29926
## [19]  -33.86290  -33.86290  -33.86290  -33.86290  -33.86290  -33.86290
## [25]  -45.35555  -33.86290  -33.86290  -33.86290  -33.86290  -33.86290
## [31]  -33.86290  -43.41941  -33.86290  -33.86290  -43.41941  -33.86290
## [37]  -42.44305  -42.44305  -46.53073  -41.99715
##
## $lower
##  [1] -302.94327 -302.94327 -302.94327 -302.94327 -273.10817 -303.91269
```

```
##  [7] -313.10292 -306.75809 -304.83940 -289.40314 -286.09304 -269.28624
## [13] -239.21919 -201.34466  -59.99307  -54.52334  -49.96932  -48.36105
## [19]  -44.57073  -42.24183  -42.24183  -42.24183  -42.24183  -42.24183
## [25]  -52.42662  -40.64123  -40.66312  -40.66312  -40.65450  -40.65450
## [31]  -40.65450  -51.80680  -42.25029  -42.25029  -52.81047  -41.43973
## [37]  -50.90850  -51.43068  -56.76423  -51.99846
##
## $upper
##  [1] -302.94327 -302.94327 -302.94327 -302.94327  150.46941  189.25763
##  [7]  198.94826  211.10166  209.18296  221.67733  218.36723  201.56043
## [13]  171.49338  105.68822  -18.56569  -18.07518  -17.75649  -24.23746
## [19]  -23.15508  -25.48398  -25.48398  -25.48398  -25.48398  -25.48398
## [25]  -38.28447  -27.08458  -27.06268  -27.06268  -27.07131  -27.07131
## [31]  -27.07131  -35.03203  -25.47552  -25.47552  -34.02835  -26.28608
## [37]  -33.97761  -33.45543  -36.29722  -31.99584
```

We also see good convergence, yet with better log marginal posterior of the best models, meaning that previous run was stuck in a local extremum.

In the Bayesian Generalized Nonlinear Model (BGNLM) analysis, you obtained the following results:

Predictor: `troot(((period*hoststar_mass)*period))` Inclusion Probability: 1.000000

This result indicates a perfect inclusion probability (1.0) for the predictor `troot(((period*hoststar_mass)*period))`, suggesting that this complex non-linear interaction term is crucial for predicting the semimajor axis.

Interpretation

1. Complex Interaction Term:

- The predictor `troot(((period*hoststar_mass)*period))` combines `period` and `hoststar_mass` in a multiplicative form, followed by a transformation.
- `troot` likely represents a specific non-linear transformation, such as a root function, which means the predictor is a transformed version of the product of `period`, `hoststar_mass`, and `period` again.

Discussion with Relation to Kepler's Third Law

1. Kepler's Third Law Kepler's Third Law states that the square of the orbital period of a planet is proportional to the cube of the semimajor axis of its orbit.

2. Implications and Interpretation

- **Alignment with Physical Laws**: The inclusion of `troot(((period*hoststar_mass)*period))` in the BGNLM model supports Kepler's Third Law by explicitly incorporating the cubic root transformation of the complicated interaction. This suggests that the model correctly captures the underlying (known in this case as Kepler's Third Law) physical relationship between the orbital period and the semimajor axis.
- **Predictive Power and Physical Meaning**: The perfect inclusion probability of this term indicates that the model effectively leverages the cubic root relationship to predict the semimajor axis. This reinforces the physical validity of the model and emphasizes the importance of incorporating both the period and the host star's mass in a non-linear fashion.

2. Comparison with Previous Models:

- **Linear and Log-Transformed Models**: These simpler models highlighted the importance of `period` and `hoststar_mass` individually, but the BGNLM model shows that their interaction, particularly in a non-linear form, is even more crucial.
- **Fractional Polynomials**: The fractional polynomial model also indicated non-linear relationships but did not capture this specific interaction and it also missed the solar mass in its functional form, highlighting the added value of BGNLM in uncovering complex dependencies.

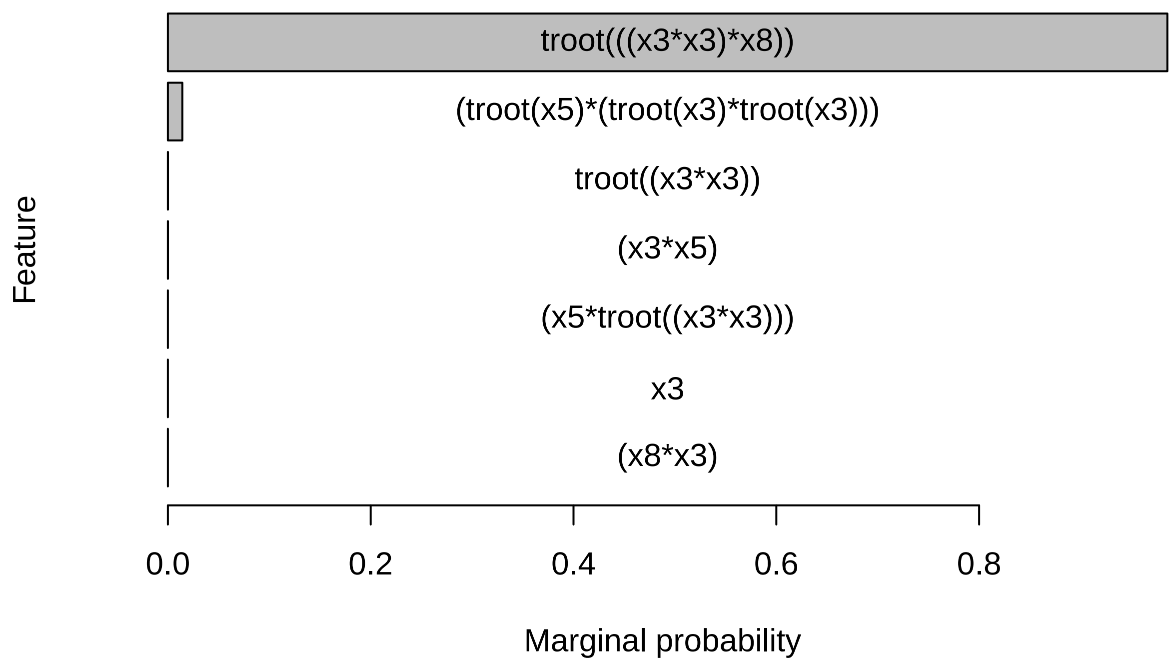
Predictions

```
preds.train.bgnlm <- predict(bgnlm, data.train[,-1])
preds.test.bgnlm <- predict(bgnlm, data.test[,-1])
r.bgnlm <- round(c(cor(data.train[,1],preds.train.bgnlm$aggr$mean)^2,cor(data.test[,1],preds.test.bgnlm$
plot(x = data.train[, 1], preds.train.bgnlm$aggr$mean, xlab = "data", ylab = "prediction", main = "Titl
points(x = data.test[, 1], preds.test.bgnlm$aggr$mean, col = "red", pch = 17)
legend("topright", legend = c(paste0("Training Data, R.sqr = ",r.bgnlm[1]), paste0("Test Data, R.sqr =
```

The predictions are expectidely perfect on both training and testing sets, as inclusion of `troot(((period*hoststar_mass)*pe` the BGNLM model effectively captures the essence of Kepler's Third Law, incorporating the orbital period and the host star's mass in a cubic root transformation. This predictor reflects the key physical relationship between these variables, demonstrating that the model is not only statistically sound but also physically meaningful. The high inclusion probability underscores the importance of this non-linear interaction in accurately predicting the semimajor axis.

And use the whole data and redo the inference.

```
library(FBMS)
transforms <- c("sin_deg","exp_dbl","p0","troot","p3")
probs <- gen.probs.gmjmcmc(transforms)
params <- gen.params.gmjmcmc(data)
set.seed(1)
bgnlm <- FBMS::fbms(semimajoraxis ~ ., data = data,transforms = transforms,runs = 24,cores = 8,P = 40,p
plot(bgnlm)
```




```
summary(bgnlm,labels = names(data.train)[-1])
```

```
##                     Importance | Feature
##                                | (troot(hoststar_mass)*(troot(period)*troot(period)))
##  ##############################| troot(((period*period)*hoststar_temperature))
##
## Best   population: 11  thread: 23  log marginal posterior: -36.98027
## Report population: 11  thread: 23  log marginal posterior: -36.98027
```

```
##                                         feats.strings marg.probs
## 1        troot(((period*period)*hoststar_temperature))  0.9855995
## 2 (troot(hoststar_mass)*(troot(period)*troot(period)))  0.0143099
```

Here we perfectly recover the true law!

We see that variation is possible in the results and that ideally maximal possible compute resources should be used to reduce the variation. But given enough resources GMJMCMC converges to being able to recover the true underlying physical law. It also seems that existing convergence tools may not be enough as they might be misleading in the situations of getting stuck in a good mode.