Package 'PowerXgammaRF'

August 21, 2026

Type Package

Title Random Forest Regression with Power Xgamma Distribution Error Model

Version 1.0.0

Description Implements Random Forest regression under the Power Xgamma distribution error model. Provides core distribution functions (density, cumulative distribution, quantile, random generation, hazard, survival), parameter estimation via Expectation-Maximization (EM) and Markov Chain Monte Carlo (MCMC), non-parametric bootstrap confidence intervals (at 90%, 95%, and 99% levels), Highest Posterior Density (HPD) intervals, Heidelberger and Welch's MCMC convergence diagnostic, convergence probability, model evaluation metrics (estimated values, bias, mean squared error, risk value), homoscedastic prediction intervals, and goodness-of-fit diagnostic tests (Kolmogorov-Smirnov and Anderson-Darling tests, Akaike Information Criterion, and Bayesian Information Criterion). References: Tyagi et al. (2022, Int. J. Stat. Reliab. Eng., 9(1), 51-60); Breiman (2001) <doi:10.1023/A:1010933404324>; Wright and Ziegler (2017) <doi:10.18637/jss.v077.i01>; Heidelberger and Welch (1983) <doi:10.1287/opre.31.6.1109>; Sen et al. (2016) <doi:10.22237/jmasm/1462076400>.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 4.0.0)

Imports ranger, coda, goftest, stats, graphics

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Language en-US

NeedsCompilation no

Author Shikhar Tyagi [aut, cre] (ORCID: <https://orcid.org/0000-0003-1606-0844>), Aruna Rajballie [aut], Vrijesh Tripathi [aut]

Maintainer Shikhar Tyagi <shikhar1093tyagi@gmail.com>

Repository CRAN

Date/Publication 2026-08-21 13:20:24 UTC

Contents

`diagnostics` *Model Diagnostics and Goodness-of-Fit Tests for Power Xgamma Error Models*

Description

Evaluates model goodness-of-fit using the Kolmogorov-Smirnov and Anderson-Darling tests, computes Information Criteria (AIC, BIC), and formats summary and diagnostic graphics.

Usage

```
ks_powerxgamma(r, eta, zeta)

ad_powerxgamma(r, eta, zeta)

## S3 method for class 'rf_powerxgamma'
print(x, ...)

## S3 method for class 'rf_powerxgamma'
summary(object, ...)

## S3 method for class 'summary.rf_powerxgamma'
print(x, ...)

## S3 method for class 'rf_powerxgamma'
plot(x, which = c(1, 2), ...)
```

Arguments

`r`	Vector of absolute residuals.
`eta`	Scale parameter $\eta > 0$.
`zeta`	Shape/power parameter $\zeta > 0$.
`x, object`	Object of class `"rf_powerxgamma"`.
`...`	Additional graphical or formatting parameters.
`which`	Integer vector indicating which plots to produce: `1` for Residual Density & Histogram, `2` for Power Xgamma Q-Q Plot.

Details

Goodness-of-fit for the Power Xgamma distribution on absolute residuals is evaluated using the Kolmogorov-Smirnov test (`ks.test`) and the tail-sensitive Anderson-Darling test (`ad.test`). Information criteria are defined as $\mathrm{AIC} = 2k - 2\ell$ and $\mathrm{BIC} = k \log n - 2\ell$, where $k = 2$ for the Power Xgamma parameters (η, ζ).

Missing Values Handling: Missing values (`NA`, `NaN`) in input residuals `r` are omitted automatically.

Value

`ks_powerxgamma` Returns an object of class `"htest"` containing Kolmogorov-Smirnov test statistic D and associated p-value.

`ad_powerxgamma` Returns an object of class `"htest"` containing Anderson-Darling test statistic A and associated p-value.

`summary.rf_powerxgamma` Returns an object of class `"summary.rf_powerxgamma"` summarizing model performance, distribution parameters, bootstrap or HPD intervals, Heidelberger-Welch convergence diagnostics, and goodness-of-fit statistics.

References

Anderson, T. W., & Darling, D. A. (1954). A test of goodness of fit. *Journal of the American Statistical Association*, 49(268), 765–769. <doi:10.1080/01621459.1954.10501232>

Tyagi, S., Kumar, S., Pandey, A., Saha, S., & Bagariya, H. (2022). Power xgamma distribution: properties and its applications to cancer data. *International Journal of Statistics and Reliability Engineering*, 9(1), 51–60.

Examples

```
set.seed(123)
r_sample <- rpowerxgamma(50, eta = 0.5, zeta = 1.2)
ks_res <- ks_powerxgamma(r_sample, eta = 0.5, zeta = 1.2)
print(ks_res)
```

`mcmc_bayesian` *Bayesian MCMC Estimation, HPD Intervals, and Heidelberger-Welch Convergence Diagnostics*

Description

Estimates the Power Xgamma distribution parameters `eta` and `zeta` using Markov Chain Monte Carlo (MCMC) sampling, and computes Highest Posterior Density (HPD) intervals at 90

Usage

```
mcmc_pxg(
  r,
  n_iter = 5000,
  burn_in = NULL,
  thin = 2,
  init_par = NULL,
  seed = NULL
)
```

Arguments

r	Vector of absolute residuals.
n_iter	Total number of MCMC iterations (default 5000).
burn_in	Number of initial burn-in iterations to discard. If NULL or burn_in >= n_iter, defaults to floor(n_iter / 5).
thin	Thinning interval for posterior chain (default 2).
init_par	Initial parameter vector c(eta, zeta). If NULL, calculated via pxg_mle.
seed	Optional random seed for reproducible sampling.

Details

MCMC sampling is performed using an adaptive Metropolis-Hastings algorithm on the log-scale of η and ζ ($\phi_1 = \log \eta, \phi_2 = \log \zeta$) to ensure strict positivity and numerical stability. Weakly informative $\mathrm{Gamma}(0.001, 0.001)$ priors are placed on η and ζ.

Convergence Diagnostics & Uncertainty Quantification:

- **HPD Intervals:** Computed using HPDinterval at 90
- **Heidelberger and Welch's Diagnostic:** Evaluated using heidel.diag to assess stationarity and calculate relative half-width accuracy.
- **Convergence Probability:** Computed as the average Cramer-von Mises stationarity test p-value across parameter chains.

Missing Values Handling: Missing values (NA, NaN) in input residuals r are omitted automatically.

Value

A list containing:

posterior_mean Posterior mean estimates of η and ζ.

posterior_median Posterior median estimates of η and ζ.

hpd_90 Highest Posterior Density intervals at 90% level.

hpd_95 Highest Posterior Density intervals at 95% level.

hpd_99 Highest Posterior Density intervals at 99% level.

heidel_diag Heidelberger and Welch's diagnostic output from heidel.diag.

stationarity_passed Named binary vector indicating whether stationarity test was passed (1 = passed, 0 = failed).

stationarity_pvalues p-values for Cramer-von Mises stationarity test.

halfwidth_passed Named binary vector indicating whether half-width test was passed (1 = passed, 0 = failed).

convergence_probability Average convergence probability across parameter chains.

acceptance_rate Empirical acceptance rate of the Metropolis-Hastings sampler.

mcmc_chain `mcmc` object of retained posterior samples.

References

Heidelberger, P., & Welch, P. D. (1983). Simulation run length control in the presence of an initial transient. *Operations Research*, 31(6), 1109–1144. <doi:10.1287/opre.31.6.1109>

Tyagi, S., Kumar, S., Pandey, A., Saha, S., & Bagariya, H. (2022). Power xgamma distribution: properties and its applications to cancer data. *International Journal of Statistics and Reliability Engineering*, 9(1), 51–60.

Examples

```
set.seed(123)
r_sample <- rpowerxgamma(50, eta = 0.5, zeta = 1.2)
mcmc_fit <- mcmc_pxg(r_sample, n_iter = 1000, burn_in = 200)
print(mcmc_fit$posterior_mean)
print(mcmc_fit$hpd_95)
print(mcmc_fit$convergence_probability)
```

`mle_em` *Maximum Likelihood Estimation and Bootstrap Inference for Power Xgamma Distribution*

Description

Computes Maximum Likelihood Estimates (MLE) / Expectation-Maximization (EM) point estimates for the Power Xgamma distribution parameters `eta` and `zeta`, and nonparametric bootstrap confidence intervals.

Usage

```
pxg_mle(r, tol = 1e-09, max_iter = 1000)

bootstrap_ci(
  y,
  X,
  n_boot = 100,
  conf_levels = c(0.9, 0.95, 0.99),
```

```
  num.trees = 100,
  seed = NULL
)
```

Arguments

r	Vector of non-negative absolute residuals.
tol	Numerical tolerance for optimization convergence (default 1e-9).
max_iter	Maximum number of optimization iterations (default 1000).
y	Response vector for bootstrap estimation.
X	Predictor matrix or data frame.
n_boot	Number of bootstrap samples (default 100).
conf_levels	Vector of nominal confidence levels (default c(0.90, 0.95, 0.99)).
num.trees	Number of trees for ranger in bootstrap fits (default 100).
seed	Optional random seed for reproducibility.

Details

Given absolute residuals $r_i = |y_i - \hat{y}_i|$, the concentrated log-likelihood for (η, ζ) is:

$$\ell(\eta,\zeta) = 2n\log\eta - n\log\zeta - n\log(1+\eta) + \left(\frac{1}{\zeta} - 1\right)\sum_{i=1}^{n}\log r_i + \sum_{i=1}^{n}\log\left(1 + \tfrac{1}{2}\eta r_i^{2/\zeta}\right) - \eta\sum_{i=1}^{n} r_i^{1/\zeta}$$

Bootstrap Resampling: Under the EM/MLE framework, non-parametric bootstrap resampling is utilized to construct empirical confidence intervals at 90

Missing Values Handling: Missing values (NA, NaN) in input residuals r are omitted automatically using na.omit. Non-positive values are bounded away from zero by .Machine$double.eps to maintain numerical stability.

Value

pxg_mle Returns a named numeric vector of length 2 containing the MLE estimates for eta and zeta.

bootstrap_ci Returns a list containing:

- eta_ci Matrix of lower and upper bounds for eta at 90%, 95%, and 99% confidence levels.
- zeta_ci Matrix of lower and upper bounds for zeta at 90%, 95%, and 99% confidence levels.
- mse_ci Matrix of lower and upper bounds for MSE at 90%, 95%, and 99% confidence levels.
- bias_ci Matrix of lower and upper bounds for model bias at 90%, 95%, and 99% confidence levels.
- risk_ci Matrix of lower and upper bounds for empirical risk at 90%, 95%, and 99% confidence levels.
- boot_eta Numeric vector of bootstrap estimates for eta.
- boot_zeta Numeric vector of bootstrap estimates for zeta.
- boot_mse Numeric vector of bootstrap estimates for MSE.
- boot_bias Numeric vector of bootstrap estimates for bias.
- boot_risk Numeric vector of bootstrap estimates for empirical risk.

References

Tyagi, S., Kumar, S., Pandey, A., Saha, S., & Bagariya, H. (2022). Power xgamma distribution: properties and its applications to cancer data. *International Journal of Statistics and Reliability Engineering*, 9(1), 51–60.

Examples

```
set.seed(123)
r_sample <- rpowerxgamma(50, eta = 0.5, zeta = 1.2)
est_pars <- pxg_mle(r_sample)
print(est_pars)
```

`powerxgamma` *The Power Xgamma Distribution*

Description

Density, distribution function, quantile function, random generation, hazard function, and survival function for the Power Xgamma distribution with scale parameter `eta` and shape/power parameter `zeta`.

Usage

```
dpowerxgamma(x, eta, zeta, log = FALSE)

dpxg(x, eta, zeta, log = FALSE)

ppowerxgamma(q, eta, zeta, lower.tail = TRUE, log.p = FALSE)

ppxg(q, eta, zeta, lower.tail = TRUE, log.p = FALSE)

qpowerxgamma(p, eta, zeta, lower.tail = TRUE, log.p = FALSE)

qpxg(p, eta, zeta, lower.tail = TRUE, log.p = FALSE)

rpowerxgamma(n, eta, zeta)

rpxg(n, eta, zeta)

hpowerxgamma(x, eta, zeta)

hpxg(x, eta, zeta)

spowerxgamma(x, eta, zeta)

spxg(x, eta, zeta)
```

Arguments

x, q	Vector of quantiles.
eta	Scale parameter ($\eta > 0$).
zeta	Shape/power parameter ($\zeta > 0$).
log, log.p	Logical or character string (TRUE or "TRUE" / FALSE or "FALSE"). If TRUE, probabilities/densities p are returned as $\log(p)$.
lower.tail	Logical or character string (TRUE or "TRUE" / FALSE or "FALSE"). If TRUE (default), probabilities are $P[X \leq x]$, otherwise, $P[X > x]$.
p	Vector of probabilities.
n	Number of observations. If length(n) > 1, the length is taken to be the number required.

Details

The Power Xgamma distribution is a flexible continuous lifetime distribution defined on $[0, \infty)$ with scale parameter $\eta > 0$ and shape/power parameter $\zeta > 0$ (Tyagi et al., 2022).

Mathematical Formulation:

- **Probability Density Function (PDF):**

$$f(x;\eta,\zeta) = \frac{\eta^2 x^{\frac{1}{\zeta}-1}\left(1+\frac{1}{2}\eta x^{2/\zeta}\right)}{\zeta(1+\eta)} \exp\left(-\eta x^{1/\zeta}\right), \quad x \geq 0,\ \eta > 0,\ \zeta > 0$$

- **Cumulative Distribution Function (CDF):**

$$F(x;\eta,\zeta) = 1 - \frac{1+\eta+\eta x^{1/\zeta}+\frac{1}{2}\eta^2 x^{2/\zeta}}{1+\eta} \exp\left(-\eta x^{1/\zeta}\right), \quad x \geq 0$$

- **Survival Function:**

$$S(x;\eta,\zeta) = 1 - F(x;\eta,\zeta) = \frac{1+\eta+\eta x^{1/\zeta}+\frac{1}{2}\eta^2 x^{2/\zeta}}{1+\eta} \exp\left(-\eta x^{1/\zeta}\right), \quad x \geq 0$$

- **Hazard Rate Function:**

$$h(x;\eta,\zeta) = \frac{f(x;\eta,\zeta)}{S(x;\eta,\zeta)} = \frac{\eta^2 x^{\frac{1}{\zeta}-1}(2+\eta x^{2/\zeta})}{\zeta(2+2\eta+2\eta x^{1/\zeta}+\eta^2 x^{2/\zeta})}, \quad x \geq 0$$

Handling of Missing and Non-Positive Values:

- Missing values (NA, NaN) in x, q, or p are preserved and return NA_real_.
- Values where x < 0 or q < 0 return 0 for density and CDF without producing warnings.
- Values where p < 0 return 0 and p > 1 return Inf in quantile functions.

Boolean Parsing: Boolean parameters (log, log.p, lower.tail) accept standard logical values (TRUE / FALSE) or character string representations ("TRUE" / "FALSE", "T" / "F").

Value

`dpowerxgamma` **(and** `dpxg`**)** Returns a numeric vector of probability density values.

`ppowerxgamma` **(and** `ppxg`**)** Returns a numeric vector of cumulative probabilities.

`qpowerxgamma` **(and** `qpxg`**)** Returns a numeric vector of quantiles.

`rpowerxgamma` **(and** `rpxg`**)** Returns a numeric vector of `n` random deviates from the Power Xgamma distribution.

`hpowerxgamma` **(and** `hpxg`**)** Returns a numeric vector of hazard rate values.

`spowerxgamma` **(and** `spxg`**)** Returns a numeric vector of survival probabilities.

References

Tyagi, S., Kumar, S., Pandey, A., Saha, S., & Bagariya, H. (2022). Power xgamma distribution: properties and its applications to cancer data. *International Journal of Statistics and Reliability Engineering*, 9(1), 51–60.

Sen, S., Maiti, S. S., & Chandra, N. (2016). The xgamma distribution: statistical properties and application. *Journal of Modern Applied Statistical Methods*, 15(1), 774–788. <doi:10.22237/jmasm/1462076400>

Examples

```
# PDF, CDF, Survival, and Hazard evaluations
dpowerxgamma(1.5, eta = 0.5, zeta = 1.2)
ppowerxgamma(1.5, eta = 0.5, zeta = 1.2)
spowerxgamma(1.5, eta = 0.5, zeta = 1.2)
hpowerxgamma(1.5, eta = 0.5, zeta = 1.2)

# Quantiles
qpowerxgamma(c(0.90, 0.95, 0.99), eta = 0.5, zeta = 1.2)

# Random variate generation
set.seed(123)
r_samples <- rpowerxgamma(10, eta = 0.5, zeta = 1.2)
print(r_samples)

# Using character string booleans
dpowerxgamma(1.5, eta = 0.5, zeta = 1.2, log = "TRUE")
ppowerxgamma(1.5, eta = 0.5, zeta = 1.2, lower.tail = "FALSE")
```

---

predict.rf_powerxgamma

Predictions and Prediction Intervals for Power Xgamma Random Forest Models

---

Description

Generates point predictions and calibrated homoscedastic prediction intervals at specified significance levels (90

Usage

```
## S3 method for class 'rf_powerxgamma'
predict(object, newdata = NULL, conf_levels = c(0.9, 0.95, 0.99), ...)
```

Arguments

`object`	Fitted object of class `"rf_powerxgamma"`.
`newdata`	Data frame of new observations for prediction. If omitted, predictions for the training data are returned.
`conf_levels`	Vector of nominal confidence levels (default `c(0.90, 0.95, 0.99)`).
`...`	Additional arguments passed to `ranger`.

Details

For an unobserved response $Y^* = f(\mathbf{x}^*) + \varepsilon^*$, the Random Forest ensemble produces point prediction $\hat{y}^* = \hat{f}(\mathbf{x}^*)$. Under the Power Xgamma error distribution $|\varepsilon^*| \sim \mathrm{PXG}(\hat{\eta}, \hat{\zeta})$, the exact $(1-\alpha)100\%$ prediction interval is:

$$\mathrm{PI}_{1-\alpha}(\mathbf{x}^*) = \left[\hat{f}(\mathbf{x}^*) - Q_{1-\alpha}(\hat{\eta}, \hat{\zeta}),\quad \hat{f}(\mathbf{x}^*) + Q_{1-\alpha}(\hat{\eta}, \hat{\zeta})\right]$$

where $Q_{1-\alpha}(\hat{\eta}, \hat{\zeta})$ is the $(1-\alpha)$-th quantile of the Power Xgamma distribution.

Missing Values Handling: If `newdata` contains missing values in predictor columns, behavior is governed by the underlying `ranger` prediction mechanism.

Value

A data frame containing:

prediction Numeric vector of point predictions $\hat{y}^*$.

lower_90, upper_90 Lower and upper bounds for 90% prediction intervals.

lower_95, upper_95 Lower and upper bounds for 95% prediction intervals.

lower_99, upper_99 Lower and upper bounds for 99% prediction intervals.

References

Tyagi, S., Kumar, S., Pandey, A., Saha, S., & Bagariya, H. (2022). Power xgamma distribution: properties and its applications to cancer data. *International Journal of Statistics and Reliability Engineering*, 9(1), 51–60.

Examples

```
set.seed(123)
train_dat <- data.frame(
  x1 = stats::rnorm(50),
  x2 = stats::runif(50),
  y = stats::rnorm(50, mean = 10, sd = 2)
)
```

```
test_dat <- data.frame(
  x1 = stats::rnorm(10),
  x2 = stats::runif(10)
)

fit <- rf_powerxgamma(y ~ x1 + x2, data = train_dat, method = "em", num.trees = 50, n_boot = 10)
preds <- predict(fit, newdata = test_dat, conf_levels = c(0.90, 0.95, 0.99))
print(preds)
```

`rf_powerxgamma` *Random Forest Regression with Power Xgamma Distribution Error Model*

Description

Fits a Random Forest regression model and models the absolute residuals using the Power Xgamma distribution. Supports Expectation-Maximization (EM) / Maximum Likelihood Estimation with non-parametric bootstrap confidence intervals, and Bayesian Markov Chain Monte Carlo (MCMC) inference with Highest Posterior Density (HPD) intervals and Heidelberger-Welch convergence diagnostics.

Usage

```
rf_powerxgamma(
  formula,
  data,
  method = c("em", "mcmc"),
  num.trees = 500,
  n_boot = 100,
  n_iter = 5000,
  burn_in = NULL,
  thin = 2,
  conf_levels = c(0.9, 0.95, 0.99),
  na.action = stats::na.omit,
  seed = NULL,
  ...
)
```

Arguments

`formula`	Object of class `formula` specifying the regression relationship.
`data`	Data frame containing response and explanatory variables.
`method`	Estimation method: `"em"` (default, EM/MLE with bootstrap CIs) or `"mcmc"` (Bayesian MCMC with HPD intervals & Heidelberger-Welch diagnostics).
`num.trees`	Number of trees to grow in the Random Forest ensemble (default `500`).

n_boot	Number of bootstrap iterations for confidence intervals when method = "em" (default 100).
n_iter	Number of MCMC iterations when method = "mcmc" (default 5000).
burn_in	Number of MCMC burn-in iterations when method = "mcmc". If NULL, defaults to floor(n_iter / 5).
thin	MCMC thinning rate (default 2).
conf_levels	Vector of nominal confidence levels (default c(0.90, 0.95, 0.99)).
na.action	Function specifying how to handle missing values (default na.omit).
seed	Optional random seed for reproducible results.
...	Additional arguments passed to ranger.

Details

The model is formulated in two stages:

1. **Stage 1 (Random Forest Non-Parametric Fitting):** The regression surface $f(\mathbf{x}) = \mathbb{E}[Y \mid \mathbf{X} = \mathbf{x}]$ is estimated non-parametrically using a Random Forest ensemble (ranger).
2. **Stage 2 (Power Xgamma Error Modeling):** Absolute residuals $r_i = |y_i - \hat{f}(\mathbf{x}_i)|$ are modeled under the Power Xgamma distribution $\mathrm{PXG}(\eta, \zeta)$.

Estimation Methods:

- **EM / MLE Method (method = "em"):** Estimates η and ζ by maximizing the concentrated log-likelihood. Non-parametric bootstrap resampling is performed to calculate empirical confidence intervals for parameter estimates, model bias, MSE, and risk at 90
- **Bayesian MCMC Method (method = "mcmc"):** Performs adaptive Metropolis-Hastings posterior sampling, extracting HPD intervals at 90

Missing Values Handling: Handled using na.action (default na.omit). Rows containing missing values in predictor variables or the response are omitted prior to model training.

Value

An object of class "rf_powerxgamma" containing:

fitted_values Fitted response predictions $\hat{y}_i$.

residuals Model residuals $e_i = y_i - \hat{y}_i$.

abs_residuals Absolute residuals $r_i = |e_i|$.

eta Estimated scale parameter $\hat{\eta}$.

zeta Estimated shape/power parameter $\hat{\zeta}$.

bias Empirical model bias $\frac{1}{n}\sum e_i$.

mse Mean squared error $\frac{1}{n}\sum e_i^2$.

rmse Root mean squared error.

mae Mean absolute error.

risk Empirical risk value under squared error loss.

conf_levels Vector of confidence levels evaluated (90%, 95%, 99%).

method Selected estimation method (`"em"` or `"mcmc"`).

bootstrap Bootstrap estimation and confidence interval results (when `method = "em"`).

mcmc MCMC posterior results, HPD intervals, and Heidelberger-Welch convergence diagnostics (when `method = "mcmc"`).

rf_fit Underlying `ranger` regression model object.

terms Terms object from the model formula.

call Matched function call.

References

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. <doi:10.1023/A:1010933404324>

Heidelberger, P., & Welch, P. D. (1983). Simulation run length control in the presence of an initial transient. *Operations Research*, 31(6), 1109–1144. <doi:10.1287/opre.31.6.1109>

Tyagi, S., Kumar, S., Pandey, A., Saha, S., & Bagariya, H. (2022). Power xgamma distribution: properties and its applications to cancer data. *International Journal of Statistics and Reliability Engineering*, 9(1), 51–60.

Wright, M. N., & Ziegler, A. (2017). ranger: A fast implementation of random forests for high dimensional data in C++ and R. *Journal of Statistical Software*, 77(1), 1–17. <doi:10.18637/jss.v077.i01>

Examples

```
set.seed(123)
dat <- data.frame(
  x1 = stats::rnorm(60),
  x2 = stats::runif(60),
  y = stats::rnorm(60, mean = 10, sd = 2)
)

# Fit model using EM/MLE method with bootstrap CIs
fit_em <- rf_powerxgamma(y ~ x1 + x2, data = dat, method = "em", num.trees = 50, n_boot = 20)
summary(fit_em)

# Fit model using Bayesian MCMC method with HPD intervals
fit_mcmc <- rf_powerxgamma(y ~ x1 + x2, data = dat, method = "mcmc", num.trees = 50, n_iter = 500)
summary(fit_mcmc)
```

Index