

Package ‘qshap’

August 23, 2026

Type Package

Title Fast Calculation of Feature Contributions in Boosting Trees

Version 2.0.0

Date 2026-08-22

Description Computes feature-specific R-squared (R²) contributions for boosting tree models using a Shapley-value-based decomposition of the total R-squared in polynomial time. Supports models fitted with 'XGBoost', 'LightGBM', and 'CatBoost', with optimized backend-specific implementations and cached tree summaries suitable for large-scale problems. Multiple visualization tools are included for interpreting and communicating feature contributions. The methodology is described in Jiang, Zhang, and Zhang (2025) <[doi:10.48550/arXiv.2407.03515](https://doi.org/10.48550/arXiv.2407.03515)>. Optional 'CatBoost' support uses the R package 'catboost', which is not distributed on CRAN; installation instructions and released binaries are provided by the CatBoost project at <<https://catboost.ai/docs/en/concepts/r-installation>>.

License GPL (>= 2)

URL https://github.com/catstats/Q-SHAP_R

BugReports https://github.com/catstats/Q-SHAP_R/issues

Imports Rcpp (>= 1.0.14), xgboost (>= 3.1.3.1), parallel, lightgbm, viridisLite, ggplot2, scales, jsonlite, methods, progress

Suggests shiny

LinkingTo Rcpp, RcppEigen

RoxygenNote 7.3.2

Encoding UTF-8

NeedsCompilation yes

Author Steven He [aut],
Zhongli Jiang [aut, cre],
Dabao Zhang [aut]

Maintainer Zhongli Jiang <zhongli.jiang.stats@gmail.com>

Repository CRAN

Date/Publication 2026-08-23 09:10:02 UTC

Contents

as.data.frame.qshap_result	2
gazer	3
get_tree	3
loss	4
plot.qshap_rsq	5
plot_loss_heatmap	6
print.qshap_result	7
print.qshap_tree_explainer	8
print.simple_tree	8
print.tree_summary	9
qshap	9
qshap_result	11
rsq	12
summary.qshap_result	14
summary.qshap_rsq	14
summary.qshap_tree_explainer	15
Index	16

as.data.frame.qshap_result
<i>Coercion method to data.frame for qshap_result</i>

Description

Coercion method to data.frame for qshap_result

Usage

```
## S3 method for class 'qshap_result'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

x	A qshap_result object
row.names	Not used
optional	Not used
...	Additional arguments (currently unused)

Value

A data.frame with columns feature (character) and rsq (numeric), sorted by rsq in decreasing order.

gazer

*Create a QSHAP Tree Explainer***Description**

Creates an explainer object for computing feature-specific Shapley values from a trained tree ensemble model. Supports XGBoost, LightGBM, and CatBoost models.

Usage

```
gazer(model, max_depth = NULL, base_score = NULL, ...)
```

Arguments

model	A model object of class <code>xgboost</code> or <code>xgb.Booster</code> from xgboost , class <code>lgb.Booster</code> from lightgbm , or class <code>catboost.Model</code> from catboost
max_depth	Maximum depth of trees, extracted from model by default.
base_score	Base score for predictions, extracted from model by default.
...	Additional arguments, for future use

Value

A class of `qshap_tree_explainer` object containing the model information and preprocessed tree structures for fast Shapley value computation

Examples

```
library(xgboost)
set.seed(42)
n <- 100
p <- 100
X <- matrix(rnorm(n * p), nrow = n, ncol = p)
y <- X[, 1] - X[, 2] + rnorm(n, sd = 0.2)
model <- xgboost(X, y, nrounds = 15, max_depth = 2, verbosity = 0, nthread = 1)
explainer <- gazeer(model)
```

get_tree

*Return a Tree Stored by Gazer***Description**

Return a Tree Stored by Gazer

Usage

```
get_tree(explainer, tree = 1L)
```

Arguments

- explainer A Q-SHAP explainer created by `gazer()`.
- tree One-based index of the tree to return.

Value

An unclassed named list containing the stored low-level tree fields.

loss	<i>Alias for <code>qshap_loss</code></i>
------	--

Description

This is a convenience alias for `qshap_loss()` that provides a shorter function name for calculating feature-specific loss contributions.

Usage

```
loss(explainer, x, y, y_mean_ori = NULL)
```

Arguments

- explainer A `qshap_tree_explainer` object created by `gazer()`
- x Feature matrix or data frame
- y Response vector
- y_mean_ori Optional pre-computed mean of y (for efficiency)

Value

A matrix of loss contributions with dimensions (n_samples, n_features)

See Also

[qshap_loss](#)

Examples

```
library(xgboost)
set.seed(42)
n <- 100
p <- 100
X <- matrix(rnorm(n * p), nrow = n, ncol = p)
y <- X[, 1] - X[, 2] + rnorm(n, sd = 0.2)
model <- xgboost(X, y, nrounds = 15, max_depth = 2, verbosity = 0, nthread = 1)
explainer <- gazer(model)
loss_matrix <- loss(explainer, X, y)
dim(loss_matrix)
```

plot.qshap_rsq

*Plot method for qshap_rsq objects***Description**

This S3 method enables ‘plot(x, …)’ where ‘x’ is a ‘qshap_rsq’ object. It dispatches to the visualization functions in ‘vis’.

Usage

```
## S3 method for class 'qshap_rsq'
plot(
  x,
  y = NULL,
  type = c("rsq", "elbow", "cumu", "gcorr", "hist", "density", "loss", "heatmap",
    "loss_heatmap"),
  ...
)
```

Arguments

x	A ‘qshap_rsq’ object.
y	Not used.
type	Plot type: one of "rsq", "elbow", "cumu", "gcorr", "hist", "density", "loss", or "heatmap". "loss_heatmap" is retained as an alias for "heatmap".
...	Passed to the underlying visualization function.

Value

A ggplot2 object (invisibly).

plot_loss_heatmap	<i>Plot observation-level contributions</i>
-------------------	---

Description

Draws an efficient raster heatmap for selected observations, with a narrow total-contribution strip. For a local Q-SHAP result, the default display is ‘local_rsqr’: observation-level contributions to the global R-squared decomposition, with legend values formatted as percentages. The underlying values remain unchanged. The diverging fill scale is centered at zero.

Usage

```
plot_loss_heatmap(
  loss,
  global_importance = NULL,
  feature_names = NULL,
  quantity = NULL,
  samples = NULL,
  n_show = 40,
  title = NULL,
  xtitle = "Feature",
  ytitle = "Observations",
  legend_title = NULL,
  low_color = "#2166AC",
  mid_color = "white",
  high_color = "#B2182B",
  rotation = 45,
  save_name = NULL
)
```

Arguments

loss	A numeric contribution matrix with observations in rows and features in columns, or a ‘qshap_rsqr’/‘qshap_result’ object produced with ‘local = TRUE’.
global_importance	Optional numeric vector used to order features from most to least globally important. For a ‘qshap_rsqr’ object, ‘\$rsqr’ is used automatically. For a matrix, the default is ‘-colSums(loss)’ when ‘quantity = "loss"' and ‘colSums(loss)’ when ‘quantity = "local_rsqr"'.
feature_names	Optional feature labels. Defaults to ‘colnames(loss)’.
quantity	Quantity to display when ‘loss’ is a Q-SHAP result: “local_rsqr” for local contributions on the R-squared scale or “loss” for raw contributions to the change in squared loss. The default uses ‘local_rsqr’ when available and otherwise falls back to ‘loss’. For a matrix, this argument controls labels and the default feature-ordering rule; the matrix itself is displayed unchanged.

<code>samples</code>	Optional observation indices or row names/IDs to display. If 'NULL', informative observations are selected automatically from both extremes of the row sums.
<code>n_show</code>	Total number of observations selected automatically when 'samples = NULL'. Approximately half are taken from each extreme.
<code>title</code>	Optional plot title. For 'local_rsqr', the default is "Observation-level contributions to the global R^2 decomposition".
<code>xtitle</code>	X-axis title.
<code>ytitle</code>	Y-axis title.
<code>legend_title</code>	Optional fill legend title. Defaults to "Local R^2 contribution" for 'local_rsqr'.
<code>low_color</code>	Color for negative contributions.
<code>mid_color</code>	Color for zero.
<code>high_color</code>	Color for positive contributions.
<code>rotation</code>	Rotation angle for feature labels.
<code>save_name</code>	Optional output path without the '.pdf' extension.

Value

A ggplot2 object, invisibly.

<code>print.qshap_result</code>	<i>Print method for qshap_result</i>
---------------------------------	--------------------------------------

Description

Print method for qshap_result

Usage

```
## S3 method for class 'qshap_result'
print(x, n = 10, ...)
```

Arguments

<code>x</code>	A qshap_result object
<code>n</code>	Integer number of top features to display (default: 10)
<code>...</code>	Additional arguments (currently unused)

Value

The input x is returned invisibly. Called primarily for its side effect of printing a summary of the qshap_result object to the console.

```
print.qshap_tree_explainer
```

Print method for qshap_tree_explainer

Description

Print method for qshap_tree_explainer

Usage

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

Arguments

x	A qshap_tree_explainer object
...	Additional arguments (currently unused)

Value

The input x is returned invisibly. Called primarily for its side effect of printing a summary of the qshap_tree_explainer object to the console.

```
print.simple_tree
```

Print method for simple_tree

Description

Print method for simple_tree

Usage

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

Arguments

x	A simple_tree object
...	Additional arguments (currently unused)

Value

The input x is returned invisibly. Called primarily for its side effect of printing a summary of the simple_tree object to the console.

print.tree_summary	<i>Print method for tree_summary</i>
--------------------	--------------------------------------

Description

Print method for tree_summary

Usage

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

Arguments

x	A tree_summary object
...	Additional arguments (currently unused)

Value

The input x is returned invisibly. Called primarily for its side effect of printing a summary of the tree_summary object to the console.

qshap	<i>Alias for rsq</i>
-------	----------------------

Description

This is a convenience alias for rsq() that provides a shorter function name for calculating feature-specific R-squared values.

Usage

```
qshap(  
  explainer,  
  x,  
  y,  
  feature_names = NULL,  
  local = FALSE,  
  nsample = NULL,  
  sd_out = TRUE,  
  nfrac = NULL,  
  random_state = 42,  
  ncore = 1L  
)
```

Arguments

<code>explainer</code>	A <code>qshap_tree_explainer</code> object created by <code>gazer()</code>
<code>x</code>	Feature matrix or data frame with <code>n</code> samples and <code>p</code> features
<code>y</code>	Response vector of length <code>n</code>
<code>feature_names</code>	Character vector of feature names. If <code>NULL</code> , uses column names from <code>x</code> .
<code>local</code>	Logical; if <code>TRUE</code> , also returns the raw observation-level squared-loss contributions in <code>loss</code> and their normalized contributions to the global R-squared decomposition in <code>local_rsqr</code> .
<code>nsample</code>	Optional integer; number of samples to use (random subsample if less than <code>nrow(x)</code>)
<code>sd_out</code>	Logical; if <code>TRUE</code> , returns standard deviations of R-squared estimates
<code>nfrac</code>	Optional numeric in (0,1); fraction of samples to use (alternative to <code>nsample</code>)
<code>random_state</code>	Integer seed for reproducible sampling
<code>ncore</code>	Number of cores for parallel processing. Use -1 for all available cores, or a positive integer. Default is 1 (no parallelization)

Value

A `qshap_result` object; see [rsqr](#) for details.

See Also

[rsqr](#)

Examples

```
library(xgboost)
set.seed(42)
n <- 100
p <- 100
X <- matrix(rnorm(n * p), nrow = n, ncol = p)
y <- X[, 1] - X[, 2] + rnorm(n, sd = 0.2)
model <- xgboost(X, y, nrounds = 15, max_depth = 2, verbosity = 0, nthread = 1)
explainer <- gazer(model)
phi_rsqr <- qshap(explainer, X, y)
print(phi_rsqr)
```

qshap_result	<i>User-friendly constructor for qshap_result</i>
--------------	---

Description

User-friendly constructor for qshap_result

Usage

```
qshap_result(  
  rsq,  
  feature_names = NULL,  
  total_rsqr = NULL,  
  n_samples = NULL,  
  n_features = NULL,  
  loss = NULL,  
  local_rsqr = NULL  
)
```

Arguments

rsq	Numeric vector of feature-specific R-squared values
feature_names	Character vector of feature names (optional)
total_rsqr	Numeric total R-squared (sum of feature-specific values)
n_samples	Integer number of samples used
n_features	Integer number of features
loss	Optional raw observation-level squared-loss contribution matrix (n_samples x n_features).
local_rsqr	Optional matrix of observation-level contributions to the global R-squared decomposition (n_samples x n_features). Its column sums equal the feature-specific values in rsq.

Value

A validated qshap_result object

rsq

*Calculate Feature-Specific R-Squared Values***Description**

Computes feature-specific R-squared values using Q-SHAP decomposition, returning a `qshap_result` object with better formatting and additional metadata. The `qshap_result` object includes feature names, total R^2 , sample counts, and provides enhanced `print()`, `summary()`, and `as.data.frame()` methods for easier analysis.

Usage

```
rsq(
  explainer,
  x,
  y,
  feature_names = NULL,
  local = FALSE,
  nsample = NULL,
  sd_out = TRUE,
  nfrac = NULL,
  random_state = 42,
  ncore = 1L
)
```

Arguments

<code>explainer</code>	A <code>qshap_tree_explainer</code> object created by <code>gazer()</code>
<code>x</code>	Feature matrix or data frame with <code>n</code> samples and <code>p</code> features
<code>y</code>	Response vector of length <code>n</code>
<code>feature_names</code>	Character vector of feature names. If <code>NULL</code> , uses column names from <code>x</code> .
<code>local</code>	Logical; if <code>TRUE</code> , also returns the raw observation-level squared-loss contributions in <code>loss</code> and their normalized contributions to the global R-squared decomposition in <code>local_rsqr</code> .
<code>nsample</code>	Optional integer; number of samples to use (random subsample if less than <code>nrow(x)</code>)
<code>sd_out</code>	Logical; if <code>TRUE</code> , returns standard deviations of R-squared estimates
<code>nfrac</code>	Optional numeric in (0,1); fraction of samples to use (alternative to <code>nsample</code>)
<code>random_state</code>	Integer seed for reproducible sampling
<code>ncore</code>	Number of cores for parallel processing. Use -1 for all available cores, or a positive integer. Default is 1 (no parallelization)

Details

The `local_rsqr` matrix contains local contributions on the R-squared scale. It decomposes each global feature-specific `rsqr` value across observations and must not be interpreted as an observation-specific coefficient of determination.

This function provides a user-friendly interface for Q-SHAP R^2 computation:

- Automatically extracts feature names from the input data
- Returns a structured object with metadata
- Provides enhanced printing with top features displayed by default
- Includes a comprehensive `summary()` method
- Can be easily converted to a data frame with `as.data.frame()`

Value

A `qshap_result` object containing:

- `rsqr`: Numeric vector of feature-specific R^2 values
- `feature_names`: Character vector of feature names
- `total_rsqr`: Total R^2 (sum of feature-specific values)
- `n_samples`: Number of samples
- `n_features`: Number of features
- `loss`: Unchanged raw observation-level contributions to the change in squared loss (if `local=TRUE`)
- `local_rsqr`: Observation-level contributions to the global R-squared decomposition, equal to $-\text{loss} / Q_{\text{emptyset}}$, where $Q_{\emptyset} = \sum_i (y_i - \bar{y})^2$. Its column sums equal `rsqr` up to numerical tolerance (if `local=TRUE`)

See Also

[qshap_result](#)

Examples

```
library(xgboost)
set.seed(42)
n <- 100
p <- 100
X <- matrix(rnorm(n * p), nrow = n, ncol = p)
y <- X[, 1] - X[, 2] + rnorm(n, sd = 0.2)
model <- xgboost(X, y, nrounds = 15, max_depth = 2, verbosity = 0, nthread = 1)
explainer <- gazer(model)
result <- rsqr(explainer, X, y)
print(result)
```

summary.qshap_result	<i>Summary method for qshap_result</i>
----------------------	--

Description

Summary method for qshap_result

Usage

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

Arguments

object	A qshap_result object
...	Additional arguments (currently unused)

Value

The input object is returned invisibly. Called primarily for its side effect of printing a detailed summary of the qshap_result object to the console.

summary.qshap_rsq	<i>Summary method for qshap_rsq objects</i>
-------------------	---

Description

Provides a summary of the qshap_rsq object, showing the top features by R-squared contribution

Usage

```
## S3 method for class 'qshap_rsq'  
summary(object, n = 10, ...)
```

Arguments

object	A qshap_rsq object
n	Integer number of top features to display (default: 10)
...	Additional arguments (currently unused)

Value

The input object is returned invisibly. Called primarily for its side effect of printing a summary of the qshap_rsq object to the console.

`summary.qshap_tree_explainer`*Summary method for qshap_tree_explainer*

Description

Provides detailed summary information about the explainer

Usage

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

Arguments

<code>object</code>	A <code>qshap_tree_explainer</code> object
<code>...</code>	Additional arguments (currently unused)

Value

The input object is returned invisibly. Called primarily for its side effect of printing a detailed summary of the `qshap_tree_explainer` object to the console.

Index

`as.data.frame.qshap_result`, [2](#)

`gazer`, [3](#)

`get_tree`, [3](#)

`loss`, [4](#)

`plot.qshap_rsqa`, [5](#)

`plot_loss_heatmap`, [6](#)

`print.qshap_result`, [7](#)

`print.qshap_tree_explainer`, [8](#)

`print.simple_tree`, [8](#)

`print.tree_summary`, [9](#)

`qshap`, [9](#)

`qshap_loss`, [4](#)

`qshap_result`, [11](#), [13](#)

`rsqa`, [10](#), [12](#)

`summary.qshap_result`, [14](#)

`summary.qshap_rsqa`, [14](#)

`summary.qshap_tree_explainer`, [15](#)