

# Package ‘audrex’

May 7, 2026

**Type** Package

**Title** Automatic Dynamic Regression using Extreme Gradient Boosting

**Version** 3.0.0

**Maintainer** Giancarlo Vercellino <giancarlo.vercellino@gmail.com>

**Description** Dynamic regression for time series using Extreme Gradient Boosting with hyper-parameter tuning via Bayesian Optimization or Random Search.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.3.3

**Depends** R (>= 4.1)

**Imports** xgboost (>= 3.1.2.1), imputeTS (>= 3.4), graphics (>= 4.5.1), grDevices (>= 4.5.1)

**URL** [https://rpubs.com/giancarlo\\_vercellino/audrex](https://rpubs.com/giancarlo_vercellino/audrex)

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Giancarlo Vercellino [aut, cre, cph]

**Repository** CRAN

**Date/Publication** 2026-01-07 20:00:02 UTC

## Contents

|                             |          |
|-----------------------------|----------|
| audrex . . . . .            | 2        |
| bitcoin_gold_oil . . . . .  | 5        |
| climate_anomalies . . . . . | 5        |
| covid_in_europe . . . . .   | 6        |
| <b>Index</b>                | <b>7</b> |

---

|        |   |
|--------|---|
| audrex | <i>audrex: Automatic Dynamic Regression using Extreme Gradient Boosting</i> |
|--------|---|

---

### Description

Dynamic regression for time series using Extreme Gradient Boosting with hyper-parameter tuning via Bayesian Optimization or Random Search.

### Usage

```
audrex(
  data,
  n_sample = 10,
  n_search = 5,
  smoother = FALSE,
  seq_len = NULL,
  diff_threshold = 0.001,
  booster = "gbtree",
  norm = NULL,
  n_dim = NULL,
  ci = 0.8,
  min_set = 30,
  max_depth = NULL,
  eta = NULL,
  gamma = NULL,
  min_child_weight = NULL,
  subsample = NULL,
  colsample_bytree = NULL,
  lambda = NULL,
  alpha = NULL,
  n_windows = 3,
  patience = 0.1,
  nrounds = 100,
  dates = NULL,
  acq = "ucb",
  kappa = 2.576,
  eps = 0,
  kernel = list(type = "exponential", power = 2),
  seed = 42
)
```

### Arguments

|          |   |
|----------|---|
| data     | A data frame with time features on columns.                                     |
| n_sample | Positive integer. Number of samples for the Bayesian Optimization. Default: 10. |

|                  |   |
|------------------|---|
| n_search         | Positive integer. Number of search steps for the Bayesian Optimization. When the parameter is set to 0, optimization is shifted to Random Search. Default: 5,   |
| smoother         | Logical. Perform optimal smoothing using standard loess. Default: FALSE   |
| seq_len          | Positive integer. Number of time-steps to be predicted. Default: NULL (automatic selection)   |
| diff_threshold   | Positive numeric. Minimum F-test threshold for differentiating each time feature (keep it low). Default: 0.001.   |
| booster          | String. Optimization methods available are: "gbtree", "gblinear". Default: "gbtree".  |
| norm             | Logical. Boolean flag to apply Yeo-Johnson normalization. Default: NULL (automatic selection from random search or bayesian search).  |
| n_dim            | Positive integer. Projection of time features in a lower dimensional space with n_dim features. The default value (NULL) sets automatically the values in c(1, n features).   |
| ci               | Confidence interval. Default: 0.8.  |
| min_set          | Positive integer. Minimum number for validation set in case of automatic resize of past dimension. Default: 30.   |
| max_depth        | Positive integer. Look to xgboost documentation for description. A vector with one or two positive integer for the search boundaries. The default value (NULL) sets automatically the values in c(1, 8).                |
| eta              | Positive numeric. Look to xgboost documentation for description. A vector with one or two positive numeric between (0, 1] for the search boundaries. The default value (NULL) sets automatically the values in c(0, 1). |
| gamma            | Positive numeric. Look to xgboost documentation for description. A vector with one or two positive numeric for the search boundaries. The default value (NULL) sets automatically the values in c(0, 100).              |
| min_child_weight | Positive numeric. Look to xgboost documentation for description. A vector with one or two positive numeric for the search boundaries. The default value (NULL) sets automatically the values in c(0, 100).              |
| subsample        | Positive numeric. Look to xgboost documentation for description. A vector with one or two positive numeric between (0, 1] for the search boundaries. The default value (NULL) sets automatically the values in c(0, 1). |
| colsample_bytree | Positive numeric. Look to xgboost documentation for description. A vector with one or two positive numeric between (0, 1] for the search boundaries. The default value (NULL) sets automatically the values in c(0, 1). |
| lambda           | Positive numeric. Look to xgboost documentation for description. A vector with one or two positive numeric for the search boundaries. The default value (NULL) sets automatically the values in c(0, 100).              |
| alpha            | Positive numeric. Look to xgboost documentation for description. A vector with one or two positive numeric for the search boundaries. The default value (NULL) sets automatically the values in c(0, 100).              |

|           |   |
|-----------|---|
| n_windows | Positive integer. Number of (expanding) windows for cross-validation. Default: 3.   |
| patience  | Positive numeric. Percentage of waiting rounds without improvement before xgboost stops. Default: 0.1   |
| nrounds   | Positive numeric. Number of round for the extreme boosting machine. Look to xgboost for description. Default: 100.                                |
| dates     | Date. Vector of dates for the time series. Default: NULL (progressive numbers).   |
| acq       | String. Parameter for Bayesian Optimization. For reference see rBayesianOptimization documentation. Default: "ucb".                               |
| kappa     | Positive numeric. Parameter for Bayesian Optimization. For reference see rBayesianOptimization documentation. Default: 2.576.                     |
| eps       | Positive numeric. Parameter for Bayesian Optimization. For reference see rBayesianOptimization documentation. Default: 0.                         |
| kernel    | List. Parameter for Bayesian Optimization. For reference see rBayesianOptimization documentation. Default: list(type = "exponential", power = 2). |
| seed      | Random seed. Default: 42.   |

### Value

This function returns a list including:

- history: a table with the models from bayesian (n\_sample + n\_search) or random search (n\_sample), their hyper-parameters and optimization metric, the weighted average rank
- models: a list with the details for each model in history
- best\_model: results for the best selected model according to the weighted average rank, including:
  - predictions: min, max, q25, q50, q75, quantile at selected ci, mean, sd, skewness and kurtosis for each time feature
  - joint\_error: max sequence error for the differentiated time features (max\_rmse, max\_mae, max\_mdae, max\_mape, max\_mase, max\_rae, max\_rse, max\_rrse, both for training and testing)
  - serie\_errors: sequence error for the differentiated time features averaged across testing windows (rmse, mae, mdae, mape, mase, rae, rse, rrse, both for training and testing)
  - pred\_stats: for each predicted time feature, IQR to range, divergence, risk ratio, upside probability, averaged across prediction time-points and at the terminal points
  - plots: a plot for each predicted time feature with highlighted median and confidence intervals
- time\_log

### Author(s)

**Maintainer:** Giancarlo Vercellino <giancarlo.vercellino@gmail.com> [copyright holder]  
 Giancarlo Vercellino <giancarlo.vercellino@gmail.com>

**See Also**

Useful links:

- [https://rpubs.com/giancarlo\\_vercellino/audrex](https://rpubs.com/giancarlo_vercellino/audrex)

**Examples**

```
audrex(covid_in_europe[, 2:5], n_samp = 3, n_search = 2, seq_len = 10) ### BAYESIAN OPTIMIZATION
audrex(covid_in_europe[, 2:5], n_samp = 5, n_search = 0, seq_len = 10) ### RANDOM SEARCH
```

---

bitcoin\_gold\_oil      *bitcoin\_gold\_oil data set*

---

**Description**

A data frame with different time series (prices and volumes) for bitcoin, gold and oil.

**Usage**

```
bitcoin_gold_oil
```

**Format**

A data frame with 18 columns and 1827 rows.

**Source**

Yahoo Finance

---

climate\_anomalies      *climate\_anomalies data set*

---

**Description**

A data frame with different two time series on global mean temperature anomalies (GMTA) and global mean sea level (GMTA).

**Usage**

```
climate_anomalies
```

**Format**

A data frame with 2 columns and 266 rows.

**Source**

Datahub.io, Climate-change collection

---

|                 |                                 |
|-----------------|---------------------------------|
| covid_in_europe | <i>covid_in_europe data set</i> |
|-----------------|---------------------------------|

---

**Description**

A data frame with with daily and cumulative cases of Covid infections and deaths in Europe since March 2021.

**Usage**

```
covid_in_europe
```

**Format**

A data frame with 5 columns and 163 rows.

**Source**

[www.ecdc.europa.eu](http://www.ecdc.europa.eu)

# Index

## \* datasets

- bitcoin\_gold\_oil, [5](#)
- climate\_anomalies, [5](#)
- covid\_in\_europe, [6](#)

audrex, [2](#)

audrex-package (audrex), [2](#)

bitcoin\_gold\_oil, [5](#)

climate\_anomalies, [5](#)

covid\_in\_europe, [6](#)