Usage

```
Arima(
  y,
  order = c(0, 0, 0),
  seasonal = c(0, 0, 0),
  xreg = NULL,
  include.mean = TRUE,
  include.drift = FALSE,
  include.constant,
  lambda = model$lambda,
  biasadj = FALSE,
  method = c("CSS-ML", "ML", "CSS"),
  model = NULL,
  x = y,
  ...
)
```

Arguments

`y`	a univariate time series of class `ts`.
`order`	A specification of the non-seasonal part of the ARIMA model: the three components (p, d, q) are the AR order, the degree of differencing, and the MA order.
`seasonal`	A specification of the seasonal part of the ARIMA model, plus the period (which defaults to frequency(y)). This should be a list with components order and period, but a specification of just a numeric vector of length 3 will be turned into a suitable list with the specification as the order.
`xreg`	Optionally, a numerical vector or matrix of external regressors, which must have the same number of rows as y. It should not be a data frame.
`include.mean`	Should the ARIMA model include a mean term? The default is `TRUE` for undifferenced series, `FALSE` for differenced ones (where a mean would not affect the fit nor predictions).
`include.drift`	Should the ARIMA model include a linear drift term? (i.e., a linear regression with ARIMA errors is fitted.) The default is `FALSE`.
`include.constant`	If `TRUE`, then `include.mean` is set to be `TRUE` for undifferenced series and `include.drift` is set to be `TRUE` for differenced series. Note that if there is more than one difference taken, no constant is included regardless of the value of this argument. This is deliberate as otherwise quadratic and higher order polynomial trends would be induced.
`lambda`	Box-Cox transformation parameter. If `lambda="auto"`, then a transformation is automatically selected using `BoxCox.lambda`. The transformation is ignored if NULL. Otherwise, data transformed before model is estimated.
`biasadj`	Use adjusted back-transformed mean for Box-Cox transformations. If transformed data is used to produce forecasts and fitted values, a regular back transformation will result in median forecasts. If biasadj is TRUE, an adjustment will be made to produce mean forecasts and fitted values.

method	Fitting method: maximum likelihood or minimize conditional sum-of-squares. The default (unless there are missing values) is to use conditional-sum-of-squares to find starting values, then maximum likelihood.
model	Output from a previous call to Arima. If model is passed, this same model is fitted to y without re-estimating any parameters.
x	Deprecated. Included for backwards compatibility.
...	Additional arguments to be passed to arima.

Details

See the arima function in the stats package.

Value

See the arima function in the stats package. The additional objects returned are

x	The time series data
xreg	The regressors used in fitting (when relevant).
sigma2	The bias adjusted MLE of the innovations variance.

Author(s)

Rob J Hyndman

See Also

auto.arima, forecast.Arima.

Examples

```
library(ggplot2)
WWWusage %>%
  Arima(order=c(3,1,0)) %>%
  forecast(h=20) %>%
  autoplot

# Fit model to first few years of AirPassengers data
air.model <- Arima(window(AirPassengers,end=1956+11/12),order=c(0,1,1),
                   seasonal=list(order=c(0,1,1),period=12),lambda=0)
plot(forecast(air.model,h=48))
lines(AirPassengers)

# Apply fitted model to later data
air.model2 <- Arima(window(AirPassengers,start=1957),model=air.model)

# Forecast accuracy measures on the log scale.
# in-sample one-step forecasts.
accuracy(air.model)
# out-of-sample one-step forecasts.
accuracy(air.model2)
```

```
# out-of-sample multi-step forecasts
accuracy(forecast(air.model,h=48,lambda=NULL),
         log(window(AirPassengers,start=1957)))
```

`arima.errors` *Errors from a regression model with ARIMA errors*

Description

Returns time series of the regression residuals from a fitted ARIMA model.

Usage

```
arima.errors(object)
```

Arguments

`object`	An object containing a time series model of class `Arima`.

Details

This is a deprecated function which is identical to `residuals.Arima(object,type="regression")` Regression residuals are equal to the original data minus the effect of any regression variables. If there are no regression variables, the errors will be identical to the original series (possibly adjusted to have zero mean).

Value

A `ts` object

Author(s)

Rob J Hyndman

See Also

`residuals.Arima`.

arimaorder *Return the order of an ARIMA or ARFIMA model*

Description

Returns the order of a univariate ARIMA or ARFIMA model.

Usage

```
arimaorder(object)
```

Arguments

object	An object of class "Arima", dQuotear or "fracdiff". Usually the result of a call to arima, Arima, auto.arima, ar, arfima or fracdiff.

Value

A numerical vector giving the values p, d and q of the ARIMA or ARFIMA model. For a seasonal ARIMA model, the returned vector contains the values p, d, q, P, D, Q and m, where m is the period of seasonality.

Author(s)

Rob J Hyndman

See Also

ar, auto.arima, Arima, arima, arfima.

Examples

```
WWWusage %>% auto.arima %>% arimaorder
```

auto.arima *Fit best ARIMA model to univariate time series*

Description

Returns best ARIMA model according to either AIC, AICc or BIC value. The function conducts a search over possible model within the order constraints provided.

Usage

```
auto.arima(
  y,
  d = NA,
  D = NA,
  max.p = 5,
  max.q = 5,
  max.P = 2,
  max.Q = 2,
  max.order = 5,
  max.d = 2,
  max.D = 1,
  start.p = 2,
  start.q = 2,
  start.P = 1,
  start.Q = 1,
  stationary = FALSE,
  seasonal = TRUE,
  ic = c("aicc", "aic", "bic"),
  stepwise = TRUE,
  nmodels = 94,
  trace = FALSE,
  approximation = (length(x) > 150 | frequency(x) > 12),
  method = NULL,
  truncate = NULL,
  xreg = NULL,
  test = c("kpss", "adf", "pp"),
  test.args = list(),
  seasonal.test = c("seas", "ocsb", "hegy", "ch"),
  seasonal.test.args = list(),
  allowdrift = TRUE,
  allowmean = TRUE,
  lambda = NULL,
  biasadj = FALSE,
  parallel = FALSE,
  num.cores = 2,
  x = y,
  ...
)
```

Arguments

y	a univariate time series
d	Order of first-differencing. If missing, will choose a value based on test.
D	Order of seasonal-differencing. If missing, will choose a value based on season.test.
max.p	Maximum value of p
max.q	Maximum value of q

max.P	Maximum value of P
max.Q	Maximum value of Q
max.order	Maximum value of p+q+P+Q if model selection is not stepwise.
max.d	Maximum number of non-seasonal differences
max.D	Maximum number of seasonal differences
start.p	Starting value of p in stepwise procedure.
start.q	Starting value of q in stepwise procedure.
start.P	Starting value of P in stepwise procedure.
start.Q	Starting value of Q in stepwise procedure.
stationary	If TRUE, restricts search to stationary models.
seasonal	If FALSE, restricts search to non-seasonal models.
ic	Information criterion to be used in model selection.
stepwise	If TRUE, will do stepwise selection (faster). Otherwise, it searches over all models. Non-stepwise selection can be very slow, especially for seasonal models.
nmodels	Maximum number of models considered in the stepwise search.
trace	If TRUE, the list of ARIMA models considered will be reported.
approximation	If TRUE, estimation is via conditional sums of squares and the information criteria used for model selection are approximated. The final model is still computed using maximum likelihood estimation. Approximation should be used for long time series or a high seasonal period to avoid excessive computation times.
method	fitting method: maximum likelihood or minimize conditional sum-of-squares. The default (unless there are missing values) is to use conditional-sum-of-squares to find starting values, then maximum likelihood. Can be abbreviated.
truncate	An integer value indicating how many observations to use in model selection. The last truncate values of the series are used to select a model when truncate is not NULL and approximation=TRUE. All observations are used if either truncate=NULL or approximation=FALSE.
xreg	Optionally, a numerical vector or matrix of external regressors, which must have the same number of rows as y. (It should not be a data frame.)
test	Type of unit root test to use. See ndiffs for details.
test.args	Additional arguments to be passed to the unit root test.
seasonal.test	This determines which method is used to select the number of seasonal differences. The default method is to use a measure of seasonal strength computed from an STL decomposition. Other possibilities involve seasonal unit root tests.
seasonal.test.args	Additional arguments to be passed to the seasonal unit root test. See nsdiffs for details.
allowdrift	If TRUE, models with drift terms are considered.
allowmean	If TRUE, models with a non-zero mean are considered.
lambda	Box-Cox transformation parameter. If lambda="auto", then a transformation is automatically selected using BoxCox.lambda. The transformation is ignored if NULL. Otherwise, data transformed before model is estimated.

biasadj	Use adjusted back-transformed mean for Box-Cox transformations. If transformed data is used to produce forecasts and fitted values, a regular back transformation will result in median forecasts. If biasadj is TRUE, an adjustment will be made to produce mean forecasts and fitted values.
parallel	If TRUE and stepwise = FALSE, then the specification search is done in parallel. This can give a significant speedup on multicore machines.
num.cores	Allows the user to specify the amount of parallel processes to be used if parallel = TRUE and stepwise = FALSE. If NULL, then the number of logical cores is automatically detected and all available cores are used.
x	Deprecated. Included for backwards compatibility.
...	Additional arguments to be passed to arima.

Details

The default arguments are designed for rapid estimation of models for many time series. If you are analysing just one time series, and can afford to take some more time, it is recommended that you set stepwise=FALSE and approximation=FALSE.

Non-stepwise selection can be slow, especially for seasonal data. The stepwise algorithm outlined in Hyndman & Khandakar (2008) is used except that the default method for selecting seasonal differences is now based on an estimate of seasonal strength (Wang, Smith & Hyndman, 2006) rather than the Canova-Hansen test. There are also some other minor variations to the algorithm described in Hyndman and Khandakar (2008).

Value

Same as for Arima

Author(s)

Rob J Hyndman

References

Hyndman, RJ and Khandakar, Y (2008) "Automatic time series forecasting: The forecast package for R", *Journal of Statistical Software*, **26**(3).

Wang, X, Smith, KA, Hyndman, RJ (2006) "Characteristic-based clustering for time series data", *Data Mining and Knowledge Discovery*, **13**(3), 335-364.

See Also

Arima

Examples

```
fit <- auto.arima(WWWusage)
plot(forecast(fit,h=20))
```