Package 'UComp'

July 6, 2021

Version 2.2.2

Title Automatic Unobserved Components Models

Description Comprehensive analysis and forecasting
of univariate time series using automatic
unobserved components models and algorithms.
Harvey, AC (1989) <doi:10.1017/CBO9781107049994>.
Pedregal, DJ and Young PC (2002) <doi:10.1002/9780470996430>.
Durbin J and Koopman SJ (2012) <doi:10.1093/acprof:oso/9780199641178.001.0001>.

Author Diego J. Pedregal [aut, cre] (<https://orcid.org/0000-0003-4958-0969>)

Depends Rcpp (>= 1.0.3), R (>= 2.10)

Imports stats

LinkingTo Rcpp, RcppArmadillo

License GPL-3

Encoding UTF-8

LazyData true

Maintainer Diego J. Pedregal <Diego.Pedregal@uclm.es>

RoxygenNote 7.1.1

Date 2021-07-05

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation yes

Repository CRAN

Date/Publication 2021-07-06 04:50:22 UTC

R topics documented:

`AIC.UComp` *AIC.UComp*

Description

Extract AIC value of UComp object

Usage

```
## S3 method for class 'UComp'
AIC(object, ..., k = 2)
```

Arguments

`object`	Object of class “UComp”.
`...`	Additional inputs to function.
`k`	The penalty per parameter to be used.

Details

Selection criteria for models with different number of parameters, the smaller AIC the better. The formula used here is $AIC = -2(ln(L) - k)/n$, where $ln(L)$ is the log-likelihood at the optimum, k is the number of parameters plus non-stationary states and n is the number of observations. Mind that this formulation differs from the usual definition that does not divide by n. This makes that AIC(m) and AIC(logLik(m)) give different results, being m an UComp object.

Author(s)

Diego J. Pedregal

See Also

`UC`, `UCmodel`, `UCvalidate`, `UCfilter`, `UCsmooth`, `UCdisturb`, `UCcomponents`

Examples

```
y <- log(AirPassengers)
m1 <- UCmodel(y, model = "llt/equal/arma(0,0)")
AIC(m1)
```

`airpas` *Airpassengers in Spain*

Description

Foreign arrivals by air in Spain in thousands of passengers (airpas).

Usage

```
airpas
```

Format

Time series objects.

Monthly data from January 1992 to December 2019

Source

airpas

Examples

```
## Not run:
 airpas

## End(Not run)
```

BIC.UComp *BIC.UComp*

Description

Extract BIC (or SBC) value of UComp object

Usage

```
## S3 method for class 'UComp'
BIC(object, ...)
```

Arguments

`object`	Object of class “UComp”.
`...`	Additional inputs to function.

Details

Selection criteria for models with different number of parameters, the smaller BIC the better. The formula used here is $BIC = (-2ln(L) + kln(n))/n$, where $ln(L)$ is the log-likelihood at the optimum, k is the number of parameters plus non-stationary states and n is the number of observations. Mind that this formulation differs from the usual definition that does not divide by n. This makes that BIC(m) and BIC(logLik(m)) give different results, being m an UComp object.

Author(s)

Diego J. Pedregal

See Also

`UC`, `UCmodel`, `UCvalidate`, `UCfilter`, `UCsmooth`, `UCdisturb`, `UCcomponents`

Examples

```
y <- log(AirPassengers)
m1 <- UCmodel(y, model = "llt/equal/arma(0,0)")
BIC(m1)
```

ch4 *Methane concentration at Cape Grim in Australia (ch4).*

Description

Methane concentration at Cape Grim in Australia (ch4).

Usage

```
ch4
```

Format

Time series objects.

Monthly data from January 1992 to December 2019

Source

CH4 data

Examples

```
## Not run:
 ch4

## End(Not run)
```

getp0 *getp0*

Description

Get initial conditions for parameters of `UComp` object

Usage

```
getp0(y, model = "llt/equal/arma(0,0)", periods = NA)
```

Arguments

`y`	a time series to forecast.
`model`	any valid `UComp` model without any ?.
`periods`	vector of fundamental period and harmonics required.

Details

Provides initial parameters of a given model for the time series. They may be changed arbitrarily by the user to include as an input `p0` to `UC` or `UCmodel` functions (see example below). There is no guarantee that the model will converge and selecting initial conditions should be used with care.

Value

A set of parameters p0 of an object of class `UComp` to use as input to `UC`, `UCmodel` or `UCsetup`.

Author(s)

Diego J. Pedregal

See Also

`UC`, `UCvalidate`, `UCfilter`, `UCsmooth`, `UCdisturb`, `UCcomponents`, `UChp`

Examples

```
p0 <- getp0(log(AirPassengers), model = "llt/equal/arma(0,0)")
p0[1] <- 0  # p0[1] <- NA
m <- UCmodel(log(AirPassengers), model = "llt/equal/arma(0,0)", p0 = p0)
```

`OECDgdp` *OECD GDP*

Description

Seasonally adjusted quarterly OECD real gross domestic product (OECDgdp).

Usage

```
OECDgdp
```

Format

Time series objects.

Quarterly data from 1962 to 2019

Source

OECDgdp

Examples

```
## Not run:
 OECDgdp

## End(Not run)
```

predict.UComp *predict.UComp*

Description

Forecasting using structural Unobseved Components models with prediction intervals

Usage

```
## S3 method for class 'UComp'
predict(object, newdata = NULL, n.ahead = NULL, level = 0.95, ...)
```

Arguments

object	Object of class “UComp”.
newdata	New output data to apply “UComp” object to.
n.ahead	Number of steps ahead to forecast or new inputs variables including their predictions.
level	Confidence level for prediction intervals.
...	Ignored.

Details

See help of UC.

Value

A matrix with the mean forecasts and lower and upper prediction intervals

Author(s)

Diego J. Pedregal

See Also

UC, UCmodel, UCvalidate, UCfilter, UCsmooth, UCdisturb, UCcomponents

Examples

```
y <- log(AirPassengers)
m1 <- UCmodel(y, model = "llt/eq/arma(0,0)")
f1 <- predict(m1)
```

`sales` *Sales index for large retailers in Spain*

Description

Sales index for food of large retailers in Spain

Usage

```
sales
```

Format

Time series objects.

Monthly data from January 1995 to December 2019

Source

sales

Examples

```
## Not run:
 sales

## End(Not run)
```

`size` *size*

Description

size of vectors or matrices

Usage

```
size(y)
```

Arguments

`y`	matrix, array or vector

Author(s)

Diego J. Pedregal

UC *UC*

Description

Runs all relevant functions for UC modelling

Usage

```
UC(
  y,
  u = NULL,
  model = "?/none/?/?",
  h = NA,
  outlier = NA,
  tTest = FALSE,
  criterion = "aic",
  periods = NA,
  verbose = FALSE,
  stepwise = FALSE,
  p0 = -9999.9,
  arma = TRUE
)
```

Arguments

`y` a time series to forecast (it may be either a numerical vector or a time series object). This is the only input required. If a vector, the additional input `periods` should be supplied compulsorily (see below).

`u` a matrix of external regressors included only in the observation equation. (it may be either a numerical vector or a time series object). If the output wanted to be forecast, matrix `u` should contain future values for inputs.

`model` the model to estimate. It is a single string indicating the type of model for each component. It allows two formats "trend/seasonal/irregular" or "trend/cycle/seasonal/irregular". The possibilities available for each component are:

- Trend: ? / none / rw / irw / llt / dt;
- Seasonal: ? / none / equal / different;
- Irregular: ? / none / arma(0, 0) / arma(p, q) - with p and q integer positive orders;
- Cycles: ? / none / combination of positive or negative numbers. Positive numbers fix the period of the cycle while negative values estimate the period taking as initial condition the absolute value of the period supplied. Several cycles with positive or negative values are possible and if a question mark is included, the model test for the existence of the cycles specified. The following are valid examples with different meanings: 48, 48?, -48, -48?, 48+60, -48+60, -48-60, 48-60, 48+60?, -48+60?, -48-60?, 48-60?.

`h` forecast horizon. If the model includes inputs h is not used, the lenght of u is used instead.

`outlier` critical level of outlier tests. If NA it does not carry out any outlier detection (default). A positive value indicates the critical minimum t test for outlier detection in any model during identification. Three types of outliers are identified, namely Additive Outliers (AO), Level Shifts (LS) and Slope Change (SC).

`tTest` augmented Dickey Fuller test for unit roots used in stepwise algorithm (TRUE / FALSE). The number of models to search for is reduced, depending on the result of this test.

`criterion` information criterion for identification ("aic", "bic" or "aicc").

`periods` vector of fundamental period and harmonics required.

`verbose` intermediate results shown about progress of estimation (TRUE / FALSE).

`stepwise` stepwise identification procedure (TRUE / FALSE).

`p0` initial parameter vector for optimisation search.

`arma` check for arma models for irregular components (TRUE / FALSE).

Details

`UC` is a function for modelling and forecasting univariate time series according to Unobserved Components models (UC). It sets up the model with a number of control variables that govern the way the rest of functions in the package work. It also estimates the model parameters by Maximum Likelihood, forecasts the data, performs smoothing, estimates model disturbances, estimates components and shows statistical diagnostics. Standard methods applicable to UComp objects are print, summary, plot, fitted, residuals, logLik, AIC, BIC, coef, predict, tsdiag.

Value

An object of class `UComp`. It is a list with fields including all the inputs and the fields listed below as outputs. All the functions in this package fill in part of the fields of any `UComp` object as specified in what follows (function `UC` fills in all of them at once):

After running `UCmodel` or `UCestim`:

- p: Estimated parameters
- v: Estimated innovations (white noise in correctly specified models)
- yFor: Forecasted values of output
- yForV: Variance of forecasted values of output
- criteria: Value of criteria for estimated model
- iter: Number of iterations in estimation
- grad: Gradient at estimated parameters
- covp: Covariance matrix of parameters

After running `UCvalidate`:

- table: Estimation and validation table

After running `UCcomponents`:

- comp: Estimated components in matrix form
- compV: Estimated components variance in matrix form

After running `UCfilter`, `UCsmooth` or `UCdisturb`:

- yFit: Fitted values of output
- yFitV: Variance of fitted values of output
- a: State estimates
- P: Variance of state estimates
- aFor: Forecasts of states
- PFor: Forecasts of states variances

After running `UCdisturb`:

- eta: State perturbations estimates
- eps: Observed perturbations estimates

Author(s)

Diego J. Pedregal

See Also

`UC`, `UCvalidate`, `UCfilter`, `UCsmooth`, `UCdisturb`, `UCcomponents`, `UChp`

Examples

```
y <- log(AirPassengers)
m1 <- UC(y)
m1 <- UC(y, model = "llt/different/arma(0,0)")
```

`UCcomponents` *UCcomponents*

Description

Estimates unobserved components of UC models Standard methods applicable to UComp objects are print, summary, plot, fitted, residuals, logLik, AIC, BIC, coef, predict, tsdiag.

Usage

```
UCcomponents(sys)
```

Arguments

`sys`	an object of type `UComp` created with `UC` or `UCmodel`

Value

The same input object with the appropriate fields filled in, in particular:

- comp: Estimated components in matrix form
- compV: Estimated components variance in matrix form

Author(s)

Diego J. Pedregal

See Also

UC, UCmodel, UCvalidate, UCfilter, UCsmooth, UCdisturb, UChp

Examples

```
m1 <- UC(log(sales))
m1 <- UCcomponents(m1)
```

UCdisturb *UCdisturb*

Description

Runs the Disturbance Smoother for UC models Standard methods applicable to UComp objects are print, summary, plot, fitted, residuals, logLik, AIC, BIC, coef, predict, tsdiag.

Usage

```
UCdisturb(sys)
```

Arguments

sys	an object of type UComp created with UC

Value

The same input object with the appropriate fields filled in, in particular:

- yFit: Fitted values of output
- yFitV: Variance of fitted values of output
- a: State estimates
- P: Variance of state estimates (diagonal of covariance matrices)
- eta: State perturbations estimates
- eps: Observed perturbations estimates

Author(s)

Diego J. Pedregal

See Also

UC, UCmodel, UCvalidate, UCfilter, UCsmooth, UCcomponents, UChp

Examples

```
m1 <- UC(log(AirPassengers))
m1 <- UCdisturb(m1)
```

UCestim *UCestim*

Description

Estimates and forecasts UC models

Usage

```
UCestim(sys)
```

Arguments

sys	an object of type UComp created with UC

Details

UCestim estimates and forecasts a time series using an UC model. The optimization method is a BFGS quasi-Newton algorithm with a backtracking line search using Armijo conditions. Parameter names in output table are the following:

- Damping: Damping factor for DT trend.
- Level: Variance of level disturbance.
- Slope: Variance of slope disturbance.
- Rho(#): Damping factor of cycle #.
- Period(#): Estimated period of cycle #.
- Var(#): Variance of cycle #.
- Seas(#): Seasonal harmonic with period #.
- Irregular: Variance of irregular component.
- AR(#): AR parameter of lag #.
- MA(#): MA parameter of lag #.
- AO#: Additive outlier in observation #.
- LS#: Level shift outlier in observation #.

- SC#: Slope change outlier in observation #.
- Beta(#): Beta parameter of input #.
- Cnst: Constant.

Standard methods applicable to UComp objects are print, summary, plot, fitted, residuals, logLik, AIC, BIC, coef, predict, tsdiag.

Value

The same input object with the appropriate fields filled in, in particular:

- p: Estimated transformed parameters
- v: Estimated innovations (white noise in correctly specified models)
- yFor: Forecast values of output
- yForV: Variance of forecast values of output
- criteria: Value of criteria for estimated model
- covp: Covariance matrix of estimated transformed parameters
- grad: Gradient of log-likelihood at the optimum
- iter: Estimation iterations

Author(s)

Diego J. Pedregal

See Also

UC, UCmodel, UCvalidate, UCfilter, UCsmooth, UCdisturb, UCcomponents, UChp

Examples

```
m1 <- UCsetup(log(AirPassengers))
m1 <- UCestim(m1)
```

UCfilter *UCfilter*

Description

Runs the Kalman Filter for UC models Standard methods applicable to UComp objects are print, summary, plot, fitted, residuals, logLik, AIC, BIC, coef, predict, tsdiag.

Usage

```
UCfilter(sys)
```

Arguments

`sys`	an object of type `UComp` created with `UC`

Value

The same input object with the appropriate fields filled in, in particular:

- yFit: Fitted values of output
- yFitV: Variance of fitted values of output
- a: State estimates
- P: Variance of state estimates (diagonal of covariance matrices)

Author(s)

Diego J. Pedregal

See Also

`UC`, `UCmodel`, `UCvalidate`, `UCsmooth`, `UCdisturb`, `UCcomponents`, `UChp`

Examples

```
m1 <- UC(log(sales))
m1 <- UCfilter(m1)
```

`UChp` *UChp*

Description

Hodrick-Prescott filter estimation

Usage

```
UChp(y, lambda = 1600)
```

Arguments

`y`	A time series object
`lambda`	Smoothing constant (default: 1600)

Value

The cycle estimation

Author(s)

Diego J. Pedregal

See Also

UC, UCmodel, UCvalidate, UCfilter, UCsmooth, UCcomponents, UCdisturb

Examples

```
cycle <- UChp(USgdp)
plot(cycle)
```

UCmodel *UCmodel*

Description

Estimates and forecasts UC general univariate models

Usage

```
UCmodel(
  y,
  u = NULL,
  model = "?/none/?/?",
  h = NA,
  outlier = NA,
  tTest = FALSE,
  criterion = "aic",
  periods = NA,
  verbose = FALSE,
  stepwise = FALSE,
  p0 = -9999.9,
  arma = TRUE
)
```

Arguments

`y` a time series to forecast (it may be either a numerical vector or a time series object). This is the only input required. If a vector, the additional input `periods` should be supplied compulsorily (see below).

`u` a matrix of external regressors included only in the observation equation. (it may be either a numerical vector or a time series object). If the output wanted to be forecast, matrix `u` should contain future values for inputs.

`model` the model to estimate. It is a single string indicating the type of model for each component. It allows two formats "trend/seasonal/irregular" or "trend/cycle/seasonal/irregular". The possibilities available for each component are:

- Trend: ? / none / rw / irw / llt / dt;
- Seasonal: ? / none / equal / different;

- Irregular: ? / none / arma(0, 0) / arma(p, q) - with p and q integer positive orders;
- Cycles: ? / none / combination of positive or negative numbers. Positive numbers fix the period of the cycle while negative values estimate the period taking as initial condition the absolute value of the period supplied. Several cycles with positive or negative values are possible and if a question mark is included, the model test for the existence of the cycles specified. The following are valid examples with different meanings: 48, 48?, -48, -48?, 48+60, -48+60, -48-60, 48-60, 48+60?, -48+60?, -48-60?, 48-60?.

`h` forecast horizon. If the model includes inputs h is not used, the lenght of u is used instead.

`outlier` critical level of outlier tests. If NA it does not carry out any outlier detection (default). A positive value indicates the critical minimum t test for outlier detection in any model during identification. Three types of outliers are identified, namely Additive Outliers (AO), Level Shifts (LS) and Slope Change (SC).

`tTest` augmented Dickey Fuller test for unit roots used in stepwise algorithm (TRUE / FALSE). The number of models to search for is reduced, depending on the result of this test.

`criterion` information criterion for identification ("aic", "bic" or "aicc").

`periods` vector of fundamental period and harmonics required.

`verbose` intermediate results shown about progress of estimation (TRUE / FALSE).

`stepwise` stepwise identification procedure (TRUE / FALSE).

`p0` initial parameter vector for optimisation search.

`arma` check for arma models for irregular components (TRUE / FALSE).

Details

`UCmodel` is a function for modelling and forecasting univariate time series according to Unobserved Components models (UC). It sets up the model with a number of control variables that govern the way the rest of functions in the package work. It also estimates the model parameters by Maximum Likelihood and forecasts the data. Standard methods applicable to UComp objects are print, summary, plot, fitted, residuals, logLik, AIC, BIC, coef, predict, tsdiag.

Value

An object of class `UComp`. It is a list with fields including all the inputs and the fields listed below as outputs. All the functions in this package fill in part of the fields of any `UComp` object as specified in what follows (function `UC` fills in all of them at once):

After running `UCmodel` or `UCestim`:

- p: Estimated parameters
- v: Estimated innovations (white noise in correctly specified models)
- yFor: Forecasted values of output
- yForV: Variance of forecasted values of output
- criteria: Value of criteria for estimated model

- iter: Number of iterations in estimation
- grad: Gradient at estimated parameters
- covp: Covariance matrix of parameters

After running `UCvalidate`:

- table: Estimation and validation table

After running `UCcomponents`:

- comp: Estimated components in matrix form
- compV: Estimated components variance in matrix form

After running `UCfilter`, `UCsmooth` or `UCdisturb`:

- yFit: Fitted values of output
- yFitV: Variance of fitted values of output
- a: State estimates
- P: Variance of state estimates
- aFor: Forecasts of states
- PFor: Forecasts of states variances

After running `UCdisturb`:

- eta: State perturbations estimates
- eps: Observed perturbations estimates

Author(s)

Diego J. Pedregal

See Also

`UC`, `UCvalidate`, `UCfilter`, `UCsmooth`, `UCdisturb`, `UCcomponents`, `UChp`

Examples

```
y <- log(AirPassengers)
m1 <- UCmodel(y)
m1 <- UCmodel(y, model = "llt/equal/arma(0,0)")
```

UComp *UComp*

Description

A package for fast automatic identification of Unobserved Components models

Details

UComp is a package for time series modelling and forecasting of Unobserved Components models inspired on the structural family due to A.C. Harvey (Basic Structural Model: BSM), enhanced with automatic identification tools by Diego J. Pedregal. The package is designed for automatic identification among a wide range of possible models for trends, cycles, seasonal and irregular components. The model may include exogenous variables. ARMA irregular components and automatic detection of outliers are also possible.

References

Harvey AC (1989). Forecasting, Structural Time Series Models and the Kalman Filter. Cam- bridge University Press.

de Jong, P. & Penzer, J. (1998). Diagnosing Shocks in Time Series, Journal of the American Statistical Association, 93, 442, 796-806.

Pedregal, D. J., & Young, P. C. (2002). Statistical approaches to modelling and forecasting time series. In M. Clements, & D. Hendry (Eds.), Companion to economic forecasting (pp. 69–104). Oxford: Blackwell Publishers.

Durbin J, Koopman SJ (2012). Time Series Analysis by State Space Methods. 38. Oxford University Press.

Proietti T. and Luati A. (2013). Maximum likelihood estimation of time series models: the Kalman filter and beyond, in Handbook of research methods and applications in empirical macroeconomics, ed. Nigar Hashimzade and Michael Thornton, E. Elgar, UK.

Maintainer

Diego J. Pedregal

Author(s)

Diego J. Pedregal

UCsetup *UCsetup*

Description

Sets up UC general univariate models

Usage

```
UCsetup(
  y,
  u = NULL,
  model = "?/none/?/?",
  h = NA,
  outlier = NA,
  tTest = FALSE,
  criterion = "aic",
  periods = NA,
  verbose = FALSE,
  stepwise = FALSE,
  p0 = -9999.9,
  arma = TRUE
)
```

Arguments

y	a time series to forecast (it may be either a numerical vector or a time series object). This is the only input required. If a vector, the additional input `periods` should be supplied compulsorily (see below).
u	a matrix of external regressors included only in the observation equation. (it may be either a numerical vector or a time series object). If the output wanted to be forecast, matrix `u` should contain future values for inputs.
model	the model to estimate. It is a single string indicating the type of model for each component. It allows two formats "trend/seasonal/irregular" or "trend/cycle/seasonal/irregular". The possibilities available for each component are: • Trend: ? / none / rw / irw / llt / dt; • Seasonal: ? / none / equal / different; • Irregular: ? / none / arma(0, 0) / arma(p, q) - with p and q integer positive orders; • Cycles: ? / none / combination of positive or negative numbers. Positive numbers fix the period of the cycle while negative values estimate the period taking as initial condition the absolute value of the period supplied. Several cycles with positive or negative values are possible and if a question mark is included, the model test for the existence of the cycles specified. The following are valid examples with different meanings: 48, 48?, -48, -48?, 48+60, -48+60, -48-60, 48-60, 48+60?, -48+60?, -48-60?, 48-60?.

`h` forecast horizon. If the model includes inputs h is not used, the lenght of u is used instead.

`outlier` critical level of outlier tests. If NA it does not carry out any outlier detection (default). A positive value indicates the critical minimum t test for outlier detection in any model during identification. Three types of outliers are identified, namely Additive Outliers (AO), Level Shifts (LS) and Slope Change (SC).

`tTest` augmented Dickey Fuller test for unit roots used in stepwise algorithm (TRUE / FALSE). The number of models to search for is reduced, depending on the result of this test.

`criterion` information criterion for identification ("aic", "bic" or "aicc").

`periods` vector of fundamental period and harmonics required.

`verbose` intermediate results shown about progress of estimation (TRUE / FALSE).

`stepwise` stepwise identification procedure (TRUE / FALSE).

`p0` initial parameter vector for optimisation search.

`arma` check for arma models for irregular components (TRUE / FALSE).

Details

See help of `UC`.

Value

An object of class `UComp`. It is a list with fields including all the inputs and the fields listed below as outputs. All the functions in this package fill in part of the fields of any `UComp` object as specified in what follows (function `UC` fills in all of them at once):

After running `UCmodel` or `UCestim`:

- p: Estimated parameters
- v: Estimated innovations (white noise in correctly specified models)
- yFor: Forecasted values of output
- yForV: Variance of forecasted values of output
- criteria: Value of criteria for estimated model
- iter: Number of iterations in estimation
- grad: Gradient at estimated parameters
- covp: Covariance matrix of parameters

After running `UCvalidate`:

- table: Estimation and validation table

After running `UCcomponents`:

- comp: Estimated components in matrix form
- compV: Estimated components variance in matrix form

After running `UCfilter`, `UCsmooth` or `UCdisturb`:

- yFit: Fitted values of output
- yFitV: Variance of fitted values of output
- a: State estimates
- P: Variance of state estimates
- aFor: Forecasts of states
- PFor: Forecasts of states variances

After running `UCdisturb`:

- eta: State perturbations estimates
- eps: Observed perturbations estimates

Standard methods applicable to UComp objects are print, summary, plot, fitted, residuals, logLik, AIC, BIC, coef, predict, tsdiag.

Author(s)

Diego J. Pedregal

See Also

`UC`, `UCmodel`, `UCvalidate`, `UCfilter`, `UCsmooth`, `UCdisturb`, `UCcomponents`, `UChp`

Examples

```
y <- log(sales)
m1 <- UCsetup(y)
m1 <- UCsetup(y, outlier = 4)
m1 <- UCsetup(y, model = "llt/equal/arma(0,0)")
m1 <- UCsetup(y, model = "?/?/?/?")
m1 <- UCsetup(y, model = "llt/?/equal/?", outlier = 4)
```

`UCsmooth` *UCsmooth*

Description

Runs the Fixed Interval Smoother for UC models Standard methods applicable to `UComp` objects are print, summary, plot, fitted, residuals, logLik, AIC, BIC, coef, predict, tsdiag.

Usage

```
UCsmooth(sys)
```

Arguments

`sys`	an object of type `UComp` created with `UC`

Value

The same input object with the appropriate fields filled in, in particular:

- yFit: Fitted values of output
- yFitV: Variance of fitted values of output
- a: State estimates
- P: Variance of state estimates (diagonal of covariance matrices)

Author(s)

Diego J. Pedregal

See Also

`UC`, `UCmodel`, `UCvalidate`, `UCfilter`, `UCdisturb`, `UCcomponents`, `UChp`

Examples

```
m1 <- UC(log(AirPassengers))
m1 <- UCsmooth(m1)
```

`UCvalidate` *UCvalidate*

Description

Shows a table of estimation and diagnostics results for UC models. Equivalent to print or summary. The table shows information in four sections: Firstly, information about the model estimated, the relevant periods of the seasonal component included, and further information about convergence. Secondly, parameters with their names are provided, the asymptotic standard errors, the ratio of the two, and the gradient at the optimum. One asterisk indicates concentrated-out parameters and two asterisks signals parameters constrained during estimation. Thirdly, information criteria and the value of the log-likelihood. Finally, diagnostic statistics about innovations, namely, the Ljung-Box Q test of absense of autocorrelation statistic for several lags, the Jarque-Bera gaussianity test, and a standard ratio of variances test.

Usage

```
UCvalidate(sys, printScreen = TRUE)
```

Arguments

`sys`	an object of type `UComp` created with `UC`
`printScreen`	print to screen or just return output table

Value

The same input object with the appropriate fields filled in, in particular:

- table: Estimation and validation table

Author(s)

Diego J. Pedregal

See Also

UC, UCmodel, UCfilter, UCsmooth, UCdisturb, UCcomponents, UChp

Examples

```
m1 <- UC(log(AirPassengers))
m1 <- UCvalidate(m1)
```

USgdp *US GDP*

Description

Seasonally adjusted quarterly US real gross domestic product (USgdp).

Usage

```
USgdp
```

Format

Time series objects.

Quarterly data from 1962 to 2019

Source

USgdp

Examples

```
## Not run:
 USgdp

## End(Not run)
```

Index