

# Package ‘ATAforecasting’

March 8, 2021

**Type** Package

**Title** Automatic Time Series Analysis and Forecasting Using the Ata Method

**Version** 0.0.52

**Date** 2021-03-05

**Description** The Ata method (Yapar et al. (2019) <doi:10.15672/hujms.461032>), an alternative to exponential smoothing (described in Yapar (2016) <doi:10.15672/HJMS.201614320580>, Yapar et al. (2017) <doi:10.15672/HJMS.2017.493>), is a new univariate time series forecasting method which provides innovative solutions to issues faced during the initialization and optimization stages of existing forecasting methods. Forecasting performance of the Ata method is superior to existing methods both in terms of easy implementation and accurate forecasting. It can be applied to non-seasonal or seasonal time series which can be decomposed into four components (remainder, level, trend and seasonal). This methodology performed well on the M3 and M4-competition data. This package was written based on Ali Sabri Taylan’s PhD dissertation.

**Maintainer** Ali Sabri Taylan <alisabritaylan@gmail.com>

**License** GPL-3

**URL** <https://github.com/alsabtay/ATAforecasting>  
<https://atamethod.wordpress.com/>

**BugReports** <https://github.com/alsabtay/ATAforecasting/issues>

**Depends** R (>= 4.0)

**Imports** graphics, forecast, Rcpp, Rdpack, seasonal, stats, stlplus,  
stR, timeSeries, TSA, tseries, utils, xts

**LinkingTo** Rcpp, RcppArmadillo

**Encoding** UTF-8

**LazyData** TRUE

**RoxygenNote** 7.1.1

**RdMacros** Rdpack

**NeedsCompilation** yes

**Author** Ali Sabri Taylan [aut, cre, cph]  
 (<<https://orcid.org/0000-0001-9514-934X>>),  
 Hanife Taylan Selamlar [aut, cph]  
 (<<https://orcid.org/0000-0002-4091-884X>>),  
 Guckan Yapar [aut, ths, cph] (<<https://orcid.org/0000-0002-0971-6676>>)

**Repository** CRAN

**Date/Publication** 2021-03-08 07:40:02 UTC

## R topics documented:

ATAforecasting-package . . . . .	2
ATA . . . . .	3
ATA.Accuracy . . . . .	9
ATA.BackTransform . . . . .	11
ATA.BoxCoxAttr . . . . .	12
ATA.CI . . . . .	13
ATA.Core . . . . .	14
ATA.Decomposition . . . . .	15
ATA.Forecast . . . . .	17
ATA.plot . . . . .	18
ATA.print . . . . .	19
ATA.SeasAttr . . . . .	19
ATA.Seasonality . . . . .	21
ATA.Shift . . . . .	23
ATA.Transform . . . . .	23
find.freq . . . . .	25
find.freq.fourier . . . . .	26
find.multi.freq . . . . .	26
fundingTR . . . . .	27
ndiffs.tseries . . . . .	27
stR_seasadj . . . . .	28
touristTR . . . . .	29
<b>Index</b>	<b>30</b>

---

ATAforecasting-package

*ATAforecasting: Automatic Time Series Analysis and Forecasting using Ata Method with Box-Cox Power Transformations Family and Seasonal Decomposition Techniques*

---

**Description**

Returns  $ATA(p,q,\phi)$  applied to  $X$ . The Ata method based on the modified simple exponential smoothing as described in Yapar, G. (2016) <doi:10.15672/HJMS.201614320580> , Yapar G., Capar, S., Selamlar, H. T., Yavuz, I. (2017) <doi:10.15672/HJMS.2017.493> and Yapar G., Selamlar, H. T., Capar, S., Yavuz, I. (2019) <doi:10.15672/hujms.461032> is a new univariate time series forecasting method which provides innovative solutions to issues faced during the initialization and optimization stages of existing methods. Forecasting performance of the Ata method is superior to existing methods both in terms of easy implementation and accurate forecasting. It can be applied to non-seasonal or seasonal time series which can be decomposed into four components (remainder, level, trend and seasonal). This methodology performed well on the M3 and M4-competition data. Returns  $ATA(p,q,\phi)$  applied to  $X$ .

**Author(s)**

Ali Sabri Taylan and Hanife Taylan Selamlar  
 Maintainer: alisabritaylan@gmail.com

ATA

*Automatic Time Series Analysis and Forecasting using Ata Method with Box-Cox Power Transformations Family and Seasonal Decomposition Techniques*

**Description**

ATA is a generic function for Ata Method forecasting. The Ata method based on the modified simple exponential smoothing as described in Yapar, G. (2016) <doi:10.15672/HJMS.201614320580> , Yapar G., Capar, S., Selamlar, H. T., Yavuz, I. (2017) <doi:10.15672/HJMS.2017.493> and Yapar G., Selamlar, H. T., Capar, S., Yavuz, I. (2019) <doi:10.15672/hujms.461032> is a new univariate time series forecasting method which provides innovative solutions to issues faced during the initialization and optimization stages of existing methods. Forecasting performance of the Ata method is superior to existing methods both in terms of easy implementation and accurate forecasting. It can be applied to non-seasonal or seasonal time series which can be decomposed into four components (remainder, level, trend and seasonal). This methodology performed well on the M3 and M4-competition data.

**Usage**

```
ATA(
  X,
  Y = NULL,
  parP = NULL,
  parQ = NULL,
  parPHI = NULL,
  model.type = NULL,
  seasonal.test = NULL,
  seasonal.model = NULL,
```

```

seasonal.period = NULL,
seasonal.type = NULL,
seasonal.test.attr = NULL,
find.period = NULL,
accuracy.type = NULL,
level.fixed = FALSE,
trend.fixed = FALSE,
trend.search = FALSE,
h = NULL,
partition.h = NULL,
holdout = FALSE,
holdout.adjustedP = TRUE,
holdin = FALSE,
transform.order = "before",
transform.method = NULL,
transform.attr = NULL,
lambda = NULL,
shift = 0,
initial.level = NULL,
initial.trend = NULL,
ci.level = 95,
start.phi = NULL,
end.phi = NULL,
size.phi = NULL,
negative.forecast = TRUE,
print.out = TRUE,
plot.out = TRUE
)

```

### Arguments

X	A numeric vector or time series of class <code>ts</code> or <code>msts</code> for in-sample.
Y	A numeric vector or time series of class <code>ts</code> or <code>msts</code> for out-sample. If you do not have out-sample data, you can split in-sample data into training and test dataset with <code>partition.h</code> argument.
parP	Value of Level parameter <code>p</code> . If <code>NULL</code> or "opt", it is estimated. <code>p</code> has all integer values from 1 to <code>length(X)</code> .
parQ	Value of Trend parameter <code>q</code> . If <code>NULL</code> or "opt", it is estimated. <code>q</code> has all integer values from 0 to <code>p</code> .
parPHI	Value of Damping Trend parameter <code>phi</code> . If <code>NULL</code> or "opt", it is estimated. <code>phi</code> has all values from 0 to 1.
model.type	An one-character string identifying method using the framework terminology. The letter "A" for additive model, the letter "M" for multiplicative model. If <code>NULL</code> , both letters will be tried and the best model (according to the accuracy measure <code>accuracy.type</code> ) returned.
seasonal.test	Testing for stationary and seasonality. If <code>TRUE</code> , the method firstly uses <code>test="adf"</code> , Augmented Dickey-Fuller, unit-root test then the test returns the least number

of differences required to pass the test at level alpha. After the unit-root test, seasonal test applies on the stationary  $X$ .

`seasonal.model` A string identifying method for seasonal decomposition. If NULL, "decomp" method is default. c("none", "decomp", "stl", "stlplus", "tbats", "stR") phrases of methods denote

- none : seasonal decomposition is not required.
- decomp : classical seasonal decomposition. If decomp, the stats package will be used.
- stl : seasonal-trend decomposition procedure based on loess developed by Cleveland et al. (1990). If stl, the stats and forecast packages will be used. Multiple seasonal periods are allowed.
- stlplus : seasonal-trend decomposition procedure based on loess developed by Cleveland et al. (1990). If stlplus, the stlplus package will be used.
- tbats : exponential smoothing state space model with Box-Cox transformation, ARMA errors, trend and seasonal components. as described in De Livera, Hyndman & Snyder (2011). Parallel processing is used by default to speed up the computations. If tbats, the forecast package will be used. Multiple seasonal periods are allowed.
- stR : seasonal-trend decomposition procedure based on regression developed by Dokumentov and Hyndman (2015). If stR, the stR package will be used. Multiple seasonal periods are allowed.
- x13 : seasonal-trend decomposition procedure based on X13ARIMA/SEATS. If x13, the seasonal package will be used.
- x11 : seasonal-trend decomposition procedure based on X11. If x11, the seasonal package will be used.

`seasonal.period`

Value(s) of seasonal periodicity. If NULL, frequency of  $X$  is default. If `seasonal.period` is not integer,  $X$  must be `msts` time series object. c(s1,s2,s3,...) for multiple period. If  $X$  has multiple periodicity, "tbats" or "stR" seasonal model have to be selected.

`seasonal.type` An one-character string identifying method for the seasonal component framework. The letter "A" for additive model, the letter "M" for multiplicative model. If NULL, both letters will be tried and the best model (according to the accuracy measure `accuracy.type`) returned. If seasonal decomposition methods except decomp with "M", Box-Cox transformation with `lambda=0` is selected.

`seasonal.test.attr`

Attributes set for unit root, seasonality tests, X13ARIMA/SEATS and X11. If NULL, `corrgram.tcrit=1.28`, `uroot.test="adf"`, `suroot.test="correlogram"`, `suroot.uroot=TRUE`, `uroot.type="trend"`, `uroot.alpha=0.05`, `suroot.alpha=0.05`, `uroot.maxd=2`, `suroot.maxD=1`, `suroot.m=frequency(X)`, `uroot.pkg="urca"`, `multi.period="min"`, `x13.estimate.maxiter=1500`, `x13.estimate.tol=1.0e-5`, `x11.estimate.maxiter=1500`, `x11.estimate.tol=1.0e-5`. If you want to change, please use `ATA.SeasAttr` function and its output. For example, you can use `seasonal.test.attr = ATA.SeasAttr(corrgram.tcrit=` equation in `ATA` function.

`find.period` Find seasonal period(s) automatically. If NULL, 0 is default. When `find.period`,

- 0 : none

	<ul style="list-style-type: none"> <li>• 1 : single period with find.freq</li> <li>• 2 : single period with forecast : : findfrequency</li> <li>• 3 : multiple period with find.freq &amp; stR</li> <li>• 4 : multiple period with find.freq &amp; tbats</li> <li>• 5 : multiple period with find.freq &amp; stl</li> </ul>
accuracy.type	<p>Accuracy measure for selection of the best model. IF NULL, sMAPE is default.</p> <ul style="list-style-type: none"> <li>• MAE : mean absolute error.</li> <li>• MSE : mean square error.</li> <li>• RMSE : root mean squared error.</li> <li>• MPE : mean percentage error.</li> <li>• MAPE : mean absolute percentage error.</li> <li>• sMAPE : symmetric mean absolute percentage error.</li> <li>• MASE : mean absolute scaled error.</li> <li>• OWA : overall weighted average of MASE and sMAPE.</li> <li>• MdAE : median absolute error.</li> <li>• MdSE : median square error.</li> <li>• RMdSE : root median squared error.</li> <li>• MdPE : median percentage error.</li> <li>• MdAPE : median absolute percentage error.</li> <li>• sMdAPE : symmetric median absolute percentage error.</li> </ul>
level.fixed	If TRUE, "pStarQ" -> First, fits ATA(p,0) where $p = p^*$ is optimized for $q=0$ . Then, fits ATA( $p^*,q$ ) where $q$ is optimized for $p = p^*$ .
trend.fixed	If TRUE, "pBullet" -> Fits ATA(p,1) where $p = p^*$ is optimized for $q = 1$ .
trend.search	If TRUE, "qBullet" -> Fits ATA(p,q) where $p = p^*$ is optimized for $q = q^*$ ( $q > 0$ ). Then, fits ATA( $p^*,q$ ) where $q$ is optimized for $p = p^*$ .
h	The number of steps to forecast ahead. When the parameter is NULL; if the frequency of X is 4 the parameter is set to 8; if the frequency of X is 12 the parameter is set to 18; the parameter is set to 6 for other cases.
partition.h	If Y is NULL, this parameter divides X into two parts: training set (in-sample) and test set (out-sample). partition.h is number of periods for forecasting and size of test set. If the value is between 0 and 1, percentage of length is active. If holdout is TRUE, this parameter will be same as h for defining holdout set.
holdout	Default is FALSE. If TRUE, ATA Method uses the holdout forecasting for accuracy measure to select the best model. In holdout forecasting, the last few data points are removed from the data series. The remaining historical data series is called in-sample data (training set), and the holdout data is called out-of-sample data (holdout set). If TRUE, partition.h will used for holdout data.
holdout.adjustedP	Default is TRUE. If TRUE, parP will be adjusted by length of training - validation sets and in-sample set when the holdout forecasting is active.
holdin	Default is FALSE. If TRUE, ATA Method uses the hold-in forecasting for accuracy measure to select the best model. In hold-in forecasting, the last h-length data points are used for accuracy measure.

<code>transform.order</code>	If "before", Box-Cox transformation family will be applied and then seasonal decomposition techniques will be applied. If "after", seasonal decomposition techniques will be applied and then Box-Cox transformation family will be applied.
<code>transform.method</code>	Transformation method → "Box_Cox", "Sqrt", "Reciprocal", "Log", "NegLog", "Modulus", "BickelDoksum", "Manly", "Dual", "YeoJohnson", "GPower", "GLog". If the transformation process needs shift parameter, <code>ATA.Transform</code> will calculate required shift parameter automatically.
<code>transform.attr</code>	Attributes set for Box-Cox transformation. If NULL, <code>bcMethod = "loglik"</code> , <code>bcLower = 0</code> , <code>bcUpper = 1</code> , <code>bcBiasAdj = FALSE</code> . If you want to change, please use <code>ATA.BoxCoxAttr</code> function and its output.
<code>lambda</code>	Box-Cox power transformation family parameter. If NULL, data transformed before model is estimated.
<code>shift</code>	Box-Cox power transformation family shifting parameter. If NULL, data transformed before model is estimated.
<code>initial.level</code>	If NULL, FALSE is default. If FALSE, ATA Method calculates the pth observation in X for level. If TRUE, ATA Method calculates average of first p value in X for level.
<code>initial.trend</code>	If NULL, FALSE is default. If FALSE, ATA Method calculates the qth observation in $X(T)-X(T-1)$ for trend. If TRUE, ATA Method calculates average of first q value in $X(T)-X(T-1)$ for trend.
<code>ci.level</code>	Confidence Interval levels for forecasting.
<code>start.phi</code>	Lower boundary for searching <code>parPHI</code> . If NULL, 0 is default.
<code>end.phi</code>	Upper boundary for searching <code>parPHI</code> . If NULL, 1 is default.
<code>size.phi</code>	Increment step for searching <code>parPHI</code> . If NULL, 0.05 is default.
<code>negative.forecast</code>	Negative values are allowed for forecasting. Default value is TRUE. If FALSE, all negative values for forecasting are set to 0.
<code>print.out</code>	Default is TRUE. If FALSE, summary of ATA Method is not shown.
<code>plot.out</code>	Default is TRUE. If FALSE, graphics of ATA Method are not shown.

### Details

Returns  $ATA(p,q,\phi)$  applied to X.

### Value

Returns an object of class `ATA`. The generic accessor functions `ATA.Forecast` and `ATA.Accuracy` extract useful features of the value returned by `ATA` and associated functions. `ATA` object is a list containing at least the following elements

- `actual` : The original time series.
- `fitted` : Fitted values (one-step forecasts). The mean of the fitted values is calculated over the ensemble.

- level : Estimated level values.
- trend : Estimated trend values.
- residuals : Original values minus fitted values.
- coefp : The weights attached to level observations.
- coefq : The weights attached to trend observations.
- p : Optimum level parameter.
- q : Optimum trend parameter.
- phi : Optimum damped trend parameter.
- model.type: Form of trend.
- h : The number of steps to forecast ahead.
- forecast : Point forecasts as a time series.
- out.sample: Test values as a time series.
- method : The name of the optimum forecasting method as a character string.
- initial.level : Selected initial level values for the time series forecasting method.
- initial.trend : Selected initial trend values for the time series forecasting method.
- level.fixed : A choice of optional level-fixed trended methods.
- trend.fixed : A choice of optional trend-fixed trended methods.
- trend.search : A choice of optional trend and level optimized trended methods if  $q > 1$ .
- transform.method : Box-Cox power transformation family method → Box\_Cox, Sqrt, Reciprocal, Log, NegLog, Modulus, BickelDoksum, Manly, Dual, YeoJohnson, GPower, GLog.
- transform.order : Define how to apply Box-Cox power transformation techniques, before or after seasonal decomposition.
- lambda : Box-Cox power transformation family parameter.
- shift : Box-Cox power transformation family shifting parameter.
- accuracy.type : Accuracy measure that is chosen for model selection.
- accuracy : In and out sample accuracy measures and its descriptives that are calculated for optimum model are given.
- holdout : Holdout forecasting is TRUE or FALSE.
- holdout.training : Training set in holdout forecasting.
- holdout.validation: Validation set in holdout forecasting.
- holdout.forecast : Holdout forecast.
- holdout.accuracy : Accuracy measure chosen for model selection in holdout forecasting.
- holdin : Hold-in forecasting is TRUE or FALSE.
- is.season : Indicates whether it contains seasonal pattern.
- seasonal.model : The name of the selected decomposition method.
- seasonal.type : Form of seasonality.
- seasonal.period : The number of seasonality periods.
- seasonal.index : Weights of seasonality.



- seasonal : Estimated seasonal values.
- seasonal.adjusted : Deseasonalized time series values.
- execution.time : The real and CPU time 'in seconds' spent by the system executing that task, including the time spent executing run-time or system services on its behalf.
- calculation.time : How much real time 'in seconds' the currently running R process has already taken.

### Author(s)

Ali Sabri Taylan and Hanife Taylan Selamlar

### References

- # Yapar G, Yavuz I, Selamlar HT (2017). "Why and How Does Exponential Smoothing Fail? An In Depth Comparison of ATA-Simple and Simple Exponential Smoothing." *Turkish Journal of Forecasting*, **1**(1), 30–39.
- # Yapar G, Capar S, Selamlar HT, Yavuz I (2018). "Modified Holt's Linear Trend Method." *Hacettepe University Journal of Mathematics and Statistics*, **47**(5), 1394–1403.
- # Yapar G (2018). "Modified simple exponential smoothing." *Hacettepe University Journal of Mathematics and Statistics*, **47**(3), 741–754.
- # Yapar G, Selamlar HT, Capar S, Yavuz I (2019). "ATA method." *Hacettepe Journal of Mathematics and Statistics*, **48**(6), 1838–1844.

### See Also

forecast, stlplus, stR, [stl](#), [decompose](#), tbats, seasadj, seasonal.

### Examples

```
demoATA <- window(touristTR, start = 2008, end = 2014.917)
ata.fit <- ATA(demoATA, h=18, seasonal.test = TRUE, seasonal.model = "stl")
ATA.plot(ATA.Forecast(ata.fit,h=18, out.sample = tail(touristTR,18)))
```

---

ATA.Accuracy

*Accuracy Measures for The ATAforecasting*

---

### Description

Returns  $ATA(p,q,\phi)$  applied to ATA object. Accuracy measures for a forecast model Returns range of summary measures of the forecast accuracy. If `out.sample` is provided, the function measures test set forecast accuracy. If `out.sample` is not provided, the function only produces training set accuracy measures. The measures calculated are:

- MAE : mean absolute error.
- MSE : mean square error.

- RMSE : root mean squared error.
- MPE : mean percentage error.
- MAPE : mean absolute percentage error.
- sMAPE : symmetric mean absolute percentage error.
- MASE : mean absolute scaled error.
- OWA : overall weighted average of MASE and sMAPE.
- MdAE : median absolute error.
- MdSE : median square error.
- RMdSE : root median squared error.
- MdPE : median percentage error.
- MdAPE : median absolute percentage error.
- sMdAPE: symmetric median absolute percentage error.

### Usage

```
ATA.Accuracy(object, out.sample = NULL)
```

### Arguments

object	An object of class ATA is required.
out.sample	A numeric vector or time series of class ts or msts for out-sample.

### Value

Matrix giving forecast accuracy measures.

### Author(s)

Ali Sabri Taylan and Hanife Taylan Selamlar

### References

#Hyndman RJ, Koehler AB (2006). "Another look at measures of forecast accuracy." *International Journal of Forecasting*, **22**(4), 679–688.

#Hyndman RJ, Athanasopoulos G (2019). *Forecasting: principles and practice*. OTexts. <https://otexts.com/fpp3/>.

### See Also

forecast, stlplus, stR, [stl](#), [decompose](#), tbats, seasadj.

### Examples

```
demoATA <- window(touristTR, start = 2008, end = 2014.917)
ata.fit <- ATA(demoATA, h=18, seasonal.test = TRUE, seasonal.model = "stl")
ata.accuracy <- ATA.Accuracy(ata.fit, tail(touristTR,18))
```

---

ATA.BackTransform      *Back Transformation Techniques for The ATAforecasting*


---

### Description

The function provides the applicability of different types of back transformation techniques for the transformed data to which the Ata method will be applied. The ATA.BackTransform function works with many different types of inputs.

### Usage

```
ATA.BackTransform(X, tMethod, tLambda, tShift, tbiasadj = FALSE, tfvar = NULL)
```

### Arguments

X	a numeric vector or time series of class <code>ts</code> or <code>msts</code> for in-sample.
tMethod	Box-Cox power transformation family is consist of "Box_Cox", "Sqrt", "Reciprocal", "Log", "NegLog", "Modulus", "BickelDoksum", "Manly", "Dual", "YeoJohnson", "GPower", "GLog" in ATAforecasting package.
tLambda	Box-Cox power transformation family parameter. If NULL, data transformed before model is estimated.
tShift	Box-Cox power transformation family shifting parameter. If NULL, data transformed before model is estimated.
tbiasadj	Use adjusted back-transformed mean for Box-Cox transformations using <code>forecast::BoxCox</code> . If transformed data is used to produce forecasts and fitted values, a regular back transformation will result in median forecasts. If <code>tbiasadj</code> is TRUE, an adjustment will be made to produce mean forecasts and fitted values.
tfvar	Optional parameter required if <code>tbiasadj=TRUE</code> . Can either be the forecast variance, or a list containing the interval level, and the corresponding upper and lower intervals.

### Value

A list object consists of transformation parameters and transformed data. ATA.Transform is a list containing at least the following elements:

- `trfmX` : Transformed data
- `tLambda` : Box-Cox power transformation family parameter
- `tShift` : Box-Cox power transformation family shifting parameter

## References

- #Tukey JW (1957). “On the Comparative Anatomy of Transformations.” *The Annals of Mathematical Statistics*, **28**(3), 602–632.
- #Box GEP, Cox DR (1964). “An Analysis of Transformations.” *Journal of the Royal Statistical Society. Series B (Methodological)*, **26**(2), 211–252.
- #Manly BFJ (1976). “Exponential data transformations.” *Journal of the Royal Statistical Society Series D*, **25**(1), 37–42.
- #John JA, Draper NR (1980). “An alternative family of transformations.” *Journal of the Royal Statistical Society Series C*, **29**(2), 190–197.
- #Bickel PJ, Doksum KA (1982). “An analysis of transformations revisited.” *Journal of the American Statistical Association*, **76**(374), 296–311.
- #Sakia RM (1992). “The Box-Cox Transformation Technique: A Review.” *Journal of the Royal Statistical Society Series D*, **41**(2), 169–178.
- #Guerrero VM (1993). “Time-series analysis supported by power transformations.” *Journal of Forecasting*, **12**(1), 37–48.
- #Yeo I, Johnson RA (2000). “A New Family of Power Transformations to Improve Normality or Symmetry.” *Biometrika*, **87**(4), 954–959.
- #Durbin BP, Hardin JS, Hawkins DM, Rocke DM (2002). “A variance-stabilizing transformation for gene-expression microarray data.” *Bioinformatics*, **18**(1), 105–110.
- #Whittaker J, Whitehead C, Somers M (2005). “The neglog transformation and quantile regression for the analysis of a large credit scoring database.” *Journal of the Royal Statistical Society Series C*, **54**(4), 863–878.
- #Yang Z (2005). “A modified family of power transformations.” *Economics Letters*, **92**(1), 14–19.
- #Kelmansky DM, Martinez EJ, Leiva V (2013). “A new variance stabilizing transformation for gene expression data analysis.” *Statistical Applications in Genetics and Molecular Biology*, **12**(6), 653–666.

---

ATA.BoxCoxAttr

*The ATA.BoxCoxAttr function works with many different types of inputs.*

---

## Description

The ATA.BoxCoxAttr function works with many different types of inputs.

## Usage

```
ATA.BoxCoxAttr(
  bcMethod = "guerrero",
  bcLower = 0,
  bcUpper = 5,
  bcBiasAdj = FALSE
)
```

**Arguments**

bcMethod	Choose method to be used in calculating lambda. "guerrero" (Guerrero, V.M. (1993) is default. Other method is "loglik").
bcLower	Lower limit for possible lambda values. The lower value is limited by -5. Default value is 0.
bcUpper	Upper limit for possible lambda values. The upper value is limited by 5. Default value is 1.
bcBiasAdj	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 bcBiasAdj is TRUE, an adjustment will be made to produce mean forecasts and fitted values. If bcBiasAdj=TRUE. Can either be the forecast variance, or a list containing the interval level, the corresponding upper and lower intervals.

**Value**

An object of class `ataattrset`.

**Author(s)**

Ali Sabri Taylan and Hanife Taylan Selamlar

**References**

- #'Box GEP, Cox DR (1964). "An Analysis of Transformations." *Journal of the Royal Statistical Society. Series B (Methodological)*, **26**(2), 211–252.
- #'Guerrero VM (1993). "Time-series analysis supported by power transformations." *Journal of Forecasting*, **12**(1), 37–48.

**See Also**

[BoxCox](#), [InvBoxCox](#), [BoxCox.lambda](#)

---

ATA.CI

*Confidence Interval function for the ATA Method*

---

**Description**

Confidence Interval function for the ATA Method

**Usage**

```
ATA.CI(object, ci.level = 95)
```

**Arguments**

<code>object</code>	An ATA object is required.
<code>ci.level</code>	Confidence level, for example: 90, 95 or 99.

**Value**

The confidence interval output for the ATA forecasts

---

ATA.Core

*The core algorithm of the ATA Method*


---

**Description**

The core algorithm of the ATA Method

**Usage**

```
ATA.Core(X, pk, qk, phik, mdlType, initialLevel, initialTrend)
```

**Arguments**

<code>X</code>	A numeric vector or time series.
<code>pk</code>	Value of Level parameter.
<code>qk</code>	Value of Trend parameter.
<code>phik</code>	Value of Damping Trend parameter.
<code>mdlType</code>	An one-character string identifying method using the framework terminology.
<code>initialLevel</code>	If NULL, FALSE is default. If FALSE, ATA Method calculates the pth observation in X for level. If TRUE, ATA Method calculates average of first p value in X for level.
<code>initialTrend</code>	If NULL, FALSE is default. If FALSE, ATA Method calculates the qth observation in $X(T)-X(T-1)$ for trend. If TRUE, ATA Method calculates average of first q value in $X(T)-X(T-1)$ for trend.

**Value**

Returns an object of class "ATA"

## Description

Automatic seasonal decomposition for ATA Method is called `ATA.Decomposition` function in `ATAforecasting` package. The function returns seasonally adjusted data constructed by removing the seasonal component. The methodology is fully automatic. The `ATA.Decomposition` function works with many different types of inputs.

## Usage

```
ATA.Decomposition(input, s.model, s.type, s.frequency, seas_attr_set)
```

## Arguments

input	It must be <code>ts</code> or <code>msts</code> or numeric object. if it is numeric object, <code>findPeriod</code> must be 1 or 2 or 3 or 4. if it is <code>msts</code> object, <code>findPeriod</code> must be 3 or 4.
s.model	<p>A string identifying method for seasonal decomposition. If <code>NULL</code>, "decomp" method is default. <code>c("none", "decomp", "stl", "stlplus", "tbats", "stR")</code> phrases of methods denote.</p> <ul style="list-style-type: none"> <li>• <code>none</code> : seasonal decomposition is not required.</li> <li>• <code>decomp</code> : classical seasonal decomposition. If <code>decomp</code>, the <code>stats</code> package will be used.</li> <li>• <code>stl</code> : seasonal-trend decomposition procedure based on loess developed by Cleveland et al. (1990). If <code>stl</code>, the <code>stats</code> and <code>forecast</code> packages will be used. Multiple seasonal periods are allowed.</li> <li>• <code>stlplus</code> : seasonal-trend decomposition procedure based on loess developed by Cleveland et al. (1990). If <code>stlplus</code>, the <code>stlplus</code> package will be used.</li> <li>• <code>tbats</code> : exponential smoothing state space model with Box–Cox transformation, ARMA errors, trend and seasonal components. as described in De Livera, Hyndman &amp; Snyder (2011). Parallel processing is used by default to speed up the computations. If <code>tbats</code>, the <code>forecast</code> package will be used. Multiple seasonal periods are allowed.</li> <li>• <code>stR</code> : seasonal-trend decomposition procedure based on regression developed by Dokumentov and Hyndman (2015). If <code>stR</code>, the <code>stR</code> package will be used. Multiple seasonal periods are allowed.</li> <li>• <code>x13</code> : seasonal-trend decomposition procedure based on X13ARIMA/SEATS. If <code>x13</code>, the <code>seasonal</code> package will be used.</li> <li>• <code>x11</code> : seasonal-trend decomposition procedure based on X11. If <code>x11</code>, the <code>seasonal</code> package will be used.</li> </ul>
s.type	A one-character string identifying method for the seasonal component framework. If <code>NULL</code> , "M" is default. The letter "A" for additive model, the letter "M" for multiplicative model.

s.frequency	Value(s) of seasonal periodicity. If s.frequency is not integer, X must be msts time series object. c(s1,s2,s3,...) for multiple period. If X has multiple periodicity, "tbats" or "stR" seasonal model have to be selected.
seas_attr_set	Assign from ATA.SeasAttr function. Attributes set for unit root and seasonality tests. For example: period of the input data which have one seasonal pattern → 12 for monthly / 4 for quarterly / 7 for daily / 5 for business days. periods of the input data which have complex/multiple seasonal patterns → c(7,354.37,365.25).

### Value

Seasonal components of the univariate time series. ATA.Decomposition is a list containing at least the following elements:

AdjustedX	Deseasonalized data
SeasIndex	Particular weights of seasonality given cycle/frequency
SeasActual	Seasonality given original data
SeasType	Seasonal decomposition technique

### Author(s)

Ali Sabri Taylan and Hanife Taylan Selamlar

### References

- # Shiskin J, Young AH, Musgrave JC (1967). "The X-11 Variant of the Census-II Method Seasonal Adjustment Program." Technical Report 15, Bureau of the U.S. Census. <https://www.census.gov/ts/papers/ShiskinYoungMusgrave1967.pdf>.
- # Dagum EB (1988). *X-11-ARIMA/88 Seasonal Adjustment Method - Foundations and Users' Manual*. Statistics Canada. <https://www.census.gov/ts/papers/Emanual.pdf>.
- # Cleveland RB, Cleveland WS, McRae JE, Terpenning I (1990). "STL: A seasonal-trend decomposition procedure based on loess." *Journal of Official Statistics*, **6**(1), 3–73.
- # Hafen RP (2010). *Local regression models: Advancements, applications, and new methods*. Ph.D. thesis, Purdue University.
- # Livera AMD, Hyndman RJ, Snyder RD (2011). "Forecasting Time Series With Complex Seasonal Patterns Using Exponential Smoothing." *Journal of the American Statistical Association*, **106**(496), 1513–1527.
- # Dokumentov A, Hyndman RJ (2015). "STR: A Seasonal-Trend Decomposition Procedure Based on Regression." Technical Report 13/15, Monash University, Department of Econometrics and Business Statistics. <https://EconPapers.repec.org/RePEc:msh:ebswps:2015-13>.
- # Dokumentov A, Hyndman RJ (2020). "STR: A Seasonal-Trend Decomposition Procedure Based on Regression."
- # Monsell BC, Aston JAD, Koopman SJ (2003). "Toward X-13?" United States Census Bureau. <https://www.census.gov/content/dam/Census/library/working-papers/2003/adrm/jsm2003bcm.pdf>.



#Monsell BC (2007). “The X-13A-S seasonal adjustment program.” In *Proceedings of the 2007 Federal Committee On Statistical Methodology Research Conference*. URL <http://www.fscm.gov/07papers/Monsell. II-B. pdf>.

#Sax C, Eddelbuettel D (2018). “Seasonal Adjustment by X-13ARIMA-SEATS in R.” *Journal of Statistical Software*, **87**(11), 1–17.

### See Also

[stl](#), [decompose](#), [seas](#), [tbats](#), [stlplus](#), [AutoSTR](#).

---

ATA.Forecast

*Forecasting Method for The ATAforecasting*


---

### Description

ATA.Forecast is a generic function for forecasting of the ATA Method.

### Usage

```
ATA.Forecast(
  object,
  h = NULL,
  out.sample = NULL,
  ci.level = 95,
  negative.forecast = TRUE
)
```

### Arguments

<code>object</code>	An ATA object is required for forecast.
<code>h</code>	Number of periods for forecasting.
<code>out.sample</code>	A numeric vector or time series of class <code>ts</code> or <code>msts</code> for out-sample.
<code>ci.level</code>	Confidence Interval levels for forecasting. Default value is 95.
<code>negative.forecast</code>	Negative values are allowed for forecasting. Default value is TRUE. If FALSE, all negative values for forecasting are set to 0.

### Value

An object of class "ATA".

### Author(s)

Ali Sabri Taylan and Hanife Taylan Selamlar

## References

- # Yapar G, Yavuz I, Selamlar HT (2017). “Why and How Does Exponential Smoothing Fail? An In Depth Comparison of ATA-Simple and Simple Exponential Smoothing.” *Turkish Journal of Forecasting*, **1**(1), 30–39.
- # Yapar G, Capar S, Selamlar HT, Yavuz I (2018). “Modified Holt’s Linear Trend Method.” *Hacettepe University Journal of Mathematics and Statistics*, **47**(5), 1394–1403.
- # Yapar G (2018). “Modified simple exponential smoothing.” *Hacettepe University Journal of Mathematics and Statistics*, **47**(3), 741–754.
- # Yapar G, Selamlar HT, Capar S, Yavuz I (2019). “ATA method.” *Hacettepe Journal of Mathematics and Statistics*, **48**(6), 1838–1844.

## See Also

forecast, stlplus, stR, [stl](#), [decompose](#), tbats, seasadj.

## Examples

```
demoATA <- window(fundingTR, start = tsp(fundingTR)[1], end = 2013)
ata.fit <- ATA(demoATA, parPHI = 1, seasonal.test = TRUE, seasonal.model = "decomp")
ATA.plot(ATA.Forecast(ata.fit, h=18))
```

---

ATA.plot

*Specialized Plot Function of The ATAforecasting*

---

## Description

Specialized Plot Function of The ATAforecasting

## Usage

```
ATA.plot(object, fcol = 4, flty = 2, flwd = 2, ...)
```

## Arguments

object	an object of ATA
fcol	line color
flty	line type
flwd	line width
...	other inputs

## Value

a graphic output for the components of the ATAforecasting

---

ATA.print

*Specialized Screen Print Function of The ATAforecasting*


---

**Description**

Specialized Screen Print Function of The ATAforecasting

**Usage**

```
ATA.print(object, ...)
```

**Arguments**

object	an object of ATA
...	other inputs

**Value**

a summary for the results of the ATAforecasting

---

ATA.SeasAttr

*Attributes Set For Unit Root and Seasonality Tests*


---

**Description**

This function is a class of seasonality tests using `corrgram.test` from `ATAforecasting` package, `ndiffs` and `nsdiffs` functions from `forecast` package. Also, this function is modified version of `ndiffs` and `nsdiffs` written by Hyndman et al. `forecast` package. Please review manual and vignette documents of latest `forecast` package. According to `forecast` package, `ndiffs` and `nsdiffs` functions to estimate the number of differences required to make a given time series stationary. `ndiffs` uses unit root tests to determine the number of differences required for time series to be made trend stationary. Several different tests are available:

- `uroot.test = 'kpss'` : the KPSS test is used with the null hypothesis that  $x$  has a stationary root against a unit-root alternative. Then the test returns the least number of differences required to pass the test at the level `uroot.alpha`.
- `uroot.test = 'adf'` : the Augmented Dickey-Fuller test is used.
- `uroot.test = 'pp'` : the Phillips-Perron test is used. In both of these cases, the null hypothesis is that  $x$  has a unit root against a stationary root alternative. Then the test returns the least number of differences required to fail the test at the level `alpha`.

`nsdiffs` uses seasonal unit root tests to determine the number of seasonal differences required for time series to be made stationary. Several different tests are available:

- `suroot.test = 'seas'` : a measure of seasonal strength is used, where differencing is selected if the seasonal strength (Wang, Smith & Hyndman, 2006) exceeds 0.64 (based on minimizing MASE when forecasting using `auto.arima` on M3 and M4 data).
- `suroot.test = 'ch'` : the Canova-Hansen (1995) test is used (with null hypothesis of deterministic seasonality)
- `suroot.test = 'hegy'` : the Hylleberg, Engle, Granger & Yoo (1990) test is used.
- `suroot.test = 'ocsb'` : the Osborn-Chui-Smith-Birchenhall (1988) test is used (with null hypothesis that a seasonal unit root exists).
- `suroot.test = 'correlogram'` : this function is written based on M4 Competition Seasonality Test.

### Usage

```
ATA.SeasAttr(
  corrgram.tcrit = 1.28,
  uroot.test = "adf",
  suroot.test = "correlogram",
  suroot.uroot = TRUE,
  uroot.type = "trend",
  uroot.alpha = 0.05,
  suroot.alpha = 0.05,
  uroot.maxd = 2,
  suroot.maxD = 1,
  suroot.m = NULL,
  uroot.pkg = "tseries",
  multi.period = "min",
  x13.estimate.maxiter = 1500,
  x13.estimate.tol = 1e-05,
  x11.estimate.maxiter = 1500,
  x11.estimate.tol = 1e-05
)
```

### Arguments

<code>corrgram.tcrit</code>	t-value for autocorrelogram.
<code>uroot.test</code>	Type of unit root test before all type seasonality test. Possible values are "adf", "pp" and "kpss".
<code>suroot.test</code>	Type of seasonal unit root test to use. Possible values are "correlogram", "seas", "hegy", "ch" and "ocsb".
<code>suroot.uroot</code>	If TRUE, unit root test for stationary before seasonal unit root test is allowed.
<code>uroot.type</code>	Specification of the deterministic component in the regression for unit root test. Possible values are "level" and "trend".
<code>uroot.alpha</code>	Significant level of the unit root test, possible values range from 0.01 to 0.1
<code>suroot.alpha</code>	Significant level of the seasonal unit root test, possible values range from 0.01 to 0.1
<code>uroot.maxd</code>	Maximum number of non-seasonal differences allowed.

<code>suroot.maxD</code>	Maximum number of seasonal differences allowed.
<code>suroot.m</code>	Deprecated. Length of seasonal period: frequency of data for <code>nsdiff</code> .
<code>uroot.pkg</code>	Using <code>urca</code> or <code>tseries</code> packages for unit root test. The default value is "urca".
<code>multi.period</code>	Selection type of multi seasonal period. min or max function for selection
<code>x13.estimate.maxiter</code>	Maximum iteration for X13ARIMA/SEATS estimation
<code>x13.estimate.tol</code>	Convergence tolerance for X13ARIMA/SEATS estimation
<code>x11.estimate.maxiter</code>	Maximum iteration for X11 estimation
<code>x11.estimate.tol</code>	Convergence tolerance for X11 estimation

**Value**

An object of class `ataattrset`.

**Author(s)**

Ali Sabri Taylan and Hanife Taylan Selamlar

**See Also**

`forecast`, `stlplus`, `stR`, [stl](#), [decompose](#), `tbats`, `seasadj`.

---

ATA.Seasonality

*Seasonality Tests for The ATAforecasting*


---

**Description**

This function is a class of seasonality tests using `corrgram.test` from `ATAforecasting` package, `ndiffs` and `nsdiffs` functions from `forecast` package. Also, this function is modified version of `ndiffs` and `nsdiffs` written by Hyndman et al. `forecast` package. Please review manual and vignette documents of latest `forecast` package. According to `forecast` package, `ndiffs` and `nsdiffs` functions to estimate the number of differences required to make a given time series stationary. `ndiffs` uses unit root tests to determine the number of differences required for time series to be made trend stationary. Several different tests are available:

- `uroot.test = 'kpss'` : the KPSS test is used with the null hypothesis that  $x$  has a stationary root against a unit-root alternative. Then the test returns the least number of differences required to pass the test at the level `uroot.alpha`.
- `uroot.test = 'adf'` : the Augmented Dickey-Fuller test is used.
- `uroot.test = 'pp'` : the Phillips-Perron test is used. In both of these cases, the null hypothesis is that  $x$  has a unit root against a stationary root alternative. Then the test returns the least number of differences required to fail the test at the level `uroot.alpha`.

nsdiffs uses seasonal unit root tests to determine the number of seasonal differences required for time series to be made stationary. Several different tests are available:

- `suroot.test = 'seas'` : a measure of seasonal strength is used, where differencing is selected if the seasonal strength (Wang, Smith & Hyndman, 2006) exceeds 0.64 (based on minimizing MASE when forecasting using `auto.arima` on M3 and M4 data).
- `suroot.test = 'ch'` : the Canova-Hansen (1995) test is used (with null hypothesis of deterministic seasonality)
- `suroot.test = 'hegy'` : the Hylleberg, Engle, Granger & Yoo (1990) test is used.
- `suroot.test = 'ocsb'` : the Osborn-Chui-Smith-Birchenhall (1988) test is used (with null hypothesis that a seasonal unit root exists).
- `suroot.test = 'correlogram'` : this function is written based on M4 Competition Seasonality Test.

### Usage

```
ATA.Seasonality(input, ppy, attr_set)
```

### Arguments

<code>input</code>	The data.
<code>ppy</code>	Frequency of the data.
<code>attr_set</code>	Assign from <code>ATA.SeasAttr</code> function. Attributes set for unit root, seasonality tests.

### Value

TRUE if the serie has seasonality. Otherwise, FALSE.

### Author(s)

Ali Sabri Taylan and Hanife Taylan Selamlar

### References

- # Dickey DA, Fuller WA (1979). "Distribution of the Estimators for Autoregressive Time Series With a Unit Root." *Journal of the American Statistical Association*, **74**(366), 427–431.
- # Said SE, Dickey DA (1984). "Testing for Unit Roots in Autoregressive-Moving Average Models of Unknown Order." *Biometrika*, **71**(3), 599–607.
- # Dickey DA, Hasza DP, Fuller WA (1984). "Testing for Unit Roots in Seasonal Time Series." *Journal of the American Statistical Association*, **79**(386), 355–367.
- # Phillips PCB, Perron P (1988). "Testing for a Unit Root in Time Series Regression." *Biometrika*, **75**(2), 335–346.
- # Osborn DR, Chui APL, Smith J, Birchenhall CR (1988). "Seasonality and the order of integration for consumption." *Oxford Bulletin of Economics and Statistics*, **50**(4), 361–377.
- # Hylleberg S, Engle RF, Granger CWJ, Yoo BS (1990). "Seasonal integration and cointegration." *Journal of Econometrics*, **1344**(1), 215–238.

# Kwiatkowski D, Phillips P, Schmidt P, Shin Y (1992). “Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root?” *Journal of Econometrics*, **54**(1–3), 159–178.

# Canova F, Hansen BE (1995). “Are Seasonal Patterns Constant over Time? A Test for Seasonal Stability.” *Journal of Business and Economic Statistics*, **13**(3), 237–252.

# 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

forecast, urca, tseries, uroot, stlplus, stR, [stl](#), [decompose](#), tbats, seasadj.

---

ATA.Shift	<i>Lag/Lead (Shift) Function for Univariate Series</i>
-----------	--

---

### Description

Lag/Lead (Shift) Function for Univariate Series

### Usage

```
ATA.Shift(x, shift_by)
```

### Arguments

x	given variable(s)
shift_by	lag or lead parameter

### Value

Generating a lag/lead variables

---

ATA.Transform	<i>Transformation Techniques for The ATAforecasting</i>
---------------	---

---

### Description

The function provides the applicability of different types of transformation techniques for the data to which the Ata method will be applied. The ATA.Transform function works with many different types of inputs.

**Usage**

```
ATA.Transform(
  X,
  tMethod = c("Box_Cox", "Sqrt", "Reciprocal", "Log", "NegLog", "Modulus",
    "BickelDoksum", "Manly", "Dual", "YeoJohnson", "GPower", "GLog"),
  tLambda,
  tShift = 0,
  bcMethod = c("loglik", "guerrero"),
  bcLower = 0,
  bcUpper = 5
)
```

**Arguments**

X	a numeric vector or time series of class <code>ts</code> or <code>msts</code> for in-sample.
tMethod	Box-Cox power transformation family is consist of "Box_Cox", "Sqrt", "Reciprocal", "Log", "NegLog", "Modulus", "BickelDoksum", "Manly", "Dual", "YeoJohnson", "GPower", "GLog" in ATAforecasting package. If the transformation process needs shift parameter, ATA.Transform will calculate required shift parameter automatically.
tLambda	Box-Cox power transformation family parameter. If NULL, data transformed before model is estimated.
tShift	Box-Cox power transformation family shifting parameter. If NULL, data transformed before model is estimated.
bcMethod	Choose method to be used in calculating lambda. "loglik" is default. Other method is "guerrero" (Guerrero, V.M. (1993)).
bcLower	Lower limit for possible lambda values. The lower value is limited by -5. Default value is 0.
bcUpper	Upper limit for possible lambda values. The upper value is limited by 5. Default value is 1.

**Value**

A list object consists of transformation parameters and transformed data. ATA.Transform is a list containing at least the following elements:

- `trfmX` : Transformed data
- `tLambda` : Box-Cox power transformation family parameter
- `tShift` : Box-Cox power transformation family shifting parameter

**References**

- #Tukey JW (1957). "On the Comparative Anatomy of Transformations." *The Annals of Mathematical Statistics*, **28**(3), 602–632.
- #Box GEP, Cox DR (1964). "An Analysis of Transformations." *Journal of the Royal Statistical Society. Series B (Methodological)*, **26**(2), 211–252.



- #'Manly BFJ (1976). "Exponential data transformations." *Journal of the Royal Statistical Society Series D*, **25**(1), 37–42.
- #'John JA, Draper NR (1980). "An alternative family of transformations." *Journal of the Royal Statistical Society Series C*, **29**(2), 190–197.
- #'Bickel PJ, Doksum KA (1982). "An analysis of transformations revisited." *Journal of the American Statistical Association*, **76**(374), 296–311.
- #'Sakia RM (1992). "The Box-Cox Transformation Technique: A Review." *Journal of the Royal Statistical Society Series D*, **41**(2), 169–178.
- #'Guerrero VM (1993). "Time-series analysis supported by power transformations." *Journal of Forecasting*, **12**(1), 37–48.
- #'Yeo I, Johnson RA (2000). "A New Family of Power Transformations to Improve Normality or Symmetry." *Biometrika*, **87**(4), 954–959.
- #'Durbin BP, Hardin JS, Hawkins DM, Rocke DM (2002). "A variance-stabilizing transformation for gene-expression microarray data." *Bioinformatics*, **18**(1), 105–110.
- #'Whittaker J, Whitehead C, Somers M (2005). "The neglog transformation and quantile regression for the analysis of a large credit scoring database." *Journal of the Royal Statistical Society Series C*, **54**(4), 863–878.
- #'Yang Z (2005). "A modified family of power transformations." *Economics Letters*, **92**(1), 14–19.
- #'Kelmansky DM, Martinez EJ, Leiva V (2013). "A new variance stabilizing transformation for gene expression data analysis." *Statistical Applications in Genetics and Molecular Biology*, **12**(6), 653–666.

---

 find.freq

 Find Frequency Using Spectral Density Of A Time Series From AR Fit
 

---

### Description

Find Frequency Using Spectral Density Of A Time Series From AR Fit

### Usage

```
find.freq(x)
```

### Arguments

x                    an univariate time series

### Value

frequency/cycle of the given time data

---

`find.freq.fourier`      *Find Frequency Using Periodogram*

---

**Description**

Find Frequency Using Periodogram

**Usage**

`find.freq.fourier(x)`

**Arguments**

`x`                      an univariate time series

**Value**

frequency/cycle of the given data

---

`find.multi.freq`      *Find Multi Frequency Using Spectral Density Of A Time Series From AR Fit*

---

**Description**

Find Multi Frequency Using Spectral Density Of A Time Series From AR Fit

**Usage**

`find.multi.freq(x)`

**Arguments**

`x`                      an univariate time series

**Value**

multi frequencies/cycles of the given data

---

`fundingTR`*Weekly Net Funding Level of Central Bank of Republic of Turkey*

---

**Description**

Weekly Net Funding Level of Central Bank of Republic of Turkey: from Jan 7, 2011 to Jan 08, 2021.

**Usage**

```
data(fundingTR)
```

**Format**

Time series data

**Source**

The Central Bank of the Republic of Turkey – CBRT.

**Examples**

```
plot(fundingTR)
```

---

`ndiffs.tseries`*Find Number of Differences Required for a Stationary Series*

---

**Description**

Number of differences required for a stationary series using `tseries` package. This function is also modified and combined version of `ndiffs` forecast and `tseries` packages. Functions to estimate the number of differences required to make a given time series stationary using `tseries` package. `ndiffs.tseries` estimates the number of first differences necessary. Please review manual and vignette documents of latest `tseries` package. `ndiffs.tseries` uses unit root tests to determine the number of differences required for time series to be made trend stationary. Several different tests are available:

- `uroot.test = 'adf'` : the Augmented Dickey-Fuller test is used.
- `uroot.test = 'pp'` : the Phillips-Perron test is used. In both of these cases, the null hypothesis is that  $x$  has a unit root against a stationary root alternative. Then the test returns the least number of differences required to fail the test at the level  $\alpha$ .
- `uroot.test = 'kpss'` : the KPSS test is used with the null hypothesis that  $x$  has a stationary root against a unit-root alternative. Then the test returns the least number of differences required to pass the test at the level  $\alpha$ .

**Usage**

```
ndiffs.tseries(x, alpha = 0.05, test = c("kpss", "adf", "pp"), max.d = 2)
```

**Arguments**

x	A univariate time series
alpha	Level of the test, possible values range from 0.01 to 0.1.
test	Type of unit root test to use
max.d	Maximum number of non-seasonal differences allowed

**Value**

An integer indicating the number of differences required for stationarity.

**Author(s)**

Ali Sabri Taylan and Hanife Taylan Selamlar

**See Also**

[ndiffs](#) [adf.test](#) [kpss.test](#) [pp.test](#)

---

stR\_seasadj

*Seasonal adjustment based on STR*

---

**Description**

This function is written to avoid conflict two functions with the same name, seasadj, in two different packages: forecast and stR.

**Usage**

```
stR_seasadj(object, include = c("Trend", "Random"))
```

**Arguments**

object	Result of AutoSTR function in ATA.Decomposition
include	Vector of component names to include in the result. The default is c("Trend", "Random").

**Value**

Seasonal adjusted data using stR package.

---

touristTR	<i>Monthly number of tourists arrived in Turkey</i>
-----------	---

---

**Description**

Monthly number of tourists arrived in Turkey: from Jan 2008 to Dec 2020.

**Usage**

```
data(touristTR)
```

**Format**

Time series data

**Source**

The Central Bank of the Republic of Turkey – CBRT.

**Examples**

```
plot(touristTR)
```

# Index

- \* **ADF**
  - ATA.Seasonality, 21
- \* **Ata**
  - ATA, 3
  - ATA.Accuracy, 9
  - ATA.BackTransform, 11
  - ATA.Decomposition, 15
  - ATA.Forecast, 17
  - ATA.Transform, 23
- \* **Bickel–Doksum**
  - ATA.BackTransform, 11
  - ATA.Transform, 23
- \* **Box–Cox**
  - ATA.BackTransform, 11
  - ATA.Transform, 23
- \* **Canova–Hansen**
  - ATA.Seasonality, 21
- \* **Guerrero**
  - ATA.BackTransform, 11
  - ATA.Transform, 23
- \* **HEGY**
  - ATA.Seasonality, 21
- \* **KPSS**
  - ATA.Seasonality, 21
- \* **Manly**
  - ATA.BackTransform, 11
  - ATA.Transform, 23
- \* **OCSB**
  - ATA.Seasonality, 21
- \* **Phillips–Perron**
  - ATA.Seasonality, 21
- \* **Yeo–Johnson**
  - ATA.BackTransform, 11
  - ATA.Transform, 23
- \* **accuracy**
  - ATA, 3
  - ATA.Accuracy, 9
  - ATA.Decomposition, 15
  - ATA.Forecast, 17
- \* **ata**
  - ATA.Seasonality, 21
- \* **correlogram**
  - ATA.Seasonality, 21
- \* **datasets**
  - fundingTR, 27
  - touristTR, 29
- \* **decomposition**
  - ATA.Decomposition, 15
  - stR\_seasadj, 28
- \* **dual**
  - ATA.BackTransform, 11
  - ATA.Transform, 23
- \* **forecast**
  - ATA, 3
  - ATA.Accuracy, 9
  - ATA.Decomposition, 15
  - ATA.Forecast, 17
- \* **glog**
  - ATA.BackTransform, 11
  - ATA.Transform, 23
- \* **gpower**
  - ATA.BackTransform, 11
  - ATA.Transform, 23
- \* **mstl**
  - ATA.Decomposition, 15
- \* **msts**
  - ATA, 3
  - ATA.Accuracy, 9
  - ATA.Decomposition, 15
  - ATA.Forecast, 17
- \* **neglog**
  - ATA.BackTransform, 11
  - ATA.Transform, 23
- \* **package**
  - ATAforecasting-package, 2
- \* **seasonal**
  - ATA.Decomposition, 15
  - ATA.Seasonality, 21

- stR\_seasadj, 28
- \* **stR**
  - stR\_seasadj, 28
- \* **stationary**
  - ndiffs.tseries, 27
- \* **transformation**
  - ATA.BackTransform, 11
  - ATA.Transform, 23
- \* **ts**
  - ATA, 3
  - ATA.Accuracy, 9
  - ATA.Decomposition, 15
  - ATA.Forecast, 17
  - ndiffs.tseries, 27
- \* **unit-root**
  - ATA.Seasonality, 21
  
- adf.test, 28
- ATA, 3
- ATA.Accuracy, 9
- ATA.BackTransform, 11
- ATA.BoxCoxAttr, 12
- ATA.CI, 13
- ATA.Core, 14
- ATA.Decomposition, 15
- ATA.Forecast, 17
- ATA.plot, 18
- ATA.print, 19
- ATA.SeasAttr, 19
- ATA.Seasonality, 21
- ATA.Shift, 23
- ATA.Transform, 23
- ATAforecasting-package, 2
- AutoSTR, 17
  
- BoxCox, 13
- BoxCox.lambda, 13
  
- decompose, 9, 10, 17, 18, 21, 23
  
- find.freq, 25
- find.freq.fourier, 26
- find.multi.freq, 26
- fundingTR, 27
  
- InvBoxCox, 13
  
- kps.test, 28
  
- ndiffs, 28
  
- ndiffs.tseries, 27
  
- pp.test, 28
  
- seas, 17
- stl, 9, 10, 17, 18, 21, 23
- stlplus, 17
- stR\_seasadj, 28
  
- tbats, 17
- touristTR, 29