

# Package ‘astsa’

March 11, 2026

**Type** Package

**Title** Applied Statistical Time Series Analysis

**Version** 2.5

**Date** 2026-03-113

**Depends** R (>= 3.5)

**Author** David Stoffer [aut, cre],  
Nicky Poison [ctb, mus, spy]

**Maintainer** David Stoffer <stoffer@pitt.edu>

**BugReports** <https://github.com/nickpoison/astsa/issues>

**Description** Contains data sets and scripts for analyzing time series in both the frequency and time domains including state space modeling as well as supporting the texts Time Series Analysis and Its Applications: With R Examples (5th ed), by R.H. Shumway and D.S. Stoffer. Springer Texts in Statistics, 2025, <[DOI:10.1007/978-3-031-70584-7](https://doi.org/10.1007/978-3-031-70584-7)>, and Time Series: A Data Analysis Approach Using R (2nd ed). Chapman-Hall, 2026, <<https://www.routledge.com/Time-Series-A-Data-Analysis-Approach-Using-R/Shumway-Stoffer/p/book/9781041031642>>. Most scripts are designed to require minimal input to produce aesthetically pleasing output for ease of use in live demonstrations and course work.

**URL** <https://dsstoffer.github.io/>, <https://nickpoison.github.io/>

**License** GPL (>= 2)

**LazyLoad** yes

**LazyData** yes

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2026-03-11 09:20:08 UTC

## Contents

astsa-package	4
acf1	6
acf2	7

acfm . . . . .	8
ar.boot . . . . .	10
ar.mcmc . . . . .	11
ar1miss . . . . .	13
arf . . . . .	14
arma.check . . . . .	14
arma.spec . . . . .	16
ARMAtoAR . . . . .	18
astsa.col . . . . .	19
autoParm . . . . .	21
autoSpec . . . . .	23
bart . . . . .	25
BCJ . . . . .	26
beamd . . . . .	27
birth . . . . .	28
blood . . . . .	28
bnrf1ebv . . . . .	29
bnrf1hvs . . . . .	30
cardox . . . . .	30
ccf2 . . . . .	31
chicken . . . . .	33
climhyd . . . . .	33
cmort . . . . .	34
cpg . . . . .	35
detrend . . . . .	35
djia . . . . .	36
dna2vector . . . . .	37
EBV . . . . .	39
econ5 . . . . .	40
EM . . . . .	41
ENSO . . . . .	43
EQ5 . . . . .	44
EQcount . . . . .	45
eqexp . . . . .	45
ESS . . . . .	47
EXP6 . . . . .	49
FDR . . . . .	49
ffbs . . . . .	50
flu . . . . .	52
fmri . . . . .	53
fmri1 . . . . .	54
gas . . . . .	55
gdp . . . . .	56
GDP23 . . . . .	56
gnp . . . . .	57
GNP23 . . . . .	58
Grid . . . . .	58
gtemp.month . . . . .	60

gtemp_both	61
gtemp_land	61
gtemp_ocean	62
Hare	63
HCT	64
hor	64
jj	65
Kfilter	66
Ksmooth	68
lag1.plot	71
lag2.plot	72
LagReg	74
lap	75
lap.xts	76
lead	78
Lynx	78
matrixpwr	79
MEI	80
MEI2	82
Months	83
mvspec	84
nyse	87
oil	87
part	88
PLT	89
polio	90
polyMul	90
pre.white	91
prodn	93
qinfl	93
qintr	94
QQnorm	95
rec	96
sales	97
salmon	97
salt	98
saltemp	99
sarima	99
sarima.for	102
sarima.sim	104
scatter.hist	106
SigExtract	107
sleep1	109
sleep2	110
so2	111
soi	112
soiltemp	112
sp500.gr	113

sp500w	114
spec.ic	114
specenv	116
speech	118
ssm	118
star	120
stoch.reg	120
sunspotz	122
SV.mcmc	123
SV.mle	125
tempr	126
test.linear	127
timex	128
trend	129
tspairs	131
tsplot	132
ttable	135
unemp	137
UnempRate	137
USpop	138
USpop20	139
varve	140
WBC	141

<b>Index</b>	<b>142</b>
--------------	------------

---

astsa-package

*Applied Statistical Time Series Analysis (more than just data)*


---

## Description

Contains data sets and scripts for analyzing time series in both the frequency and time domains including state space modeling as well as supporting the texts [Time Series Analysis and Its Applications: With R Examples \(5th ed, 2025\)](#) and [Time Series: A Data Analysis Approach Using R, \(2nd ed, 2026\)](#). Most scripts are designed to need minimal input to produce aesthetically pleasing output for ease of use in live demonstrations and in course work.

## Details

```

Package:  astsa
Type:     Package
Version:  2.5
Date:     2026-03-13
License:  GPL (>= 2)
LazyLoad: yes
LazyData: yes

```

## Warning

If the package `dplyr` is loaded, it will mask the base scripts `filter` and `lag` (among other things) that a time series analyst uses often. In this case, whenever you analyze time series data, we suggest you either:

- (1) Detach it if it's loaded but not being used:

```
detach(package:dplyr)
```

- (2) If you want to use it, fix it:

```
library(dplyr, exclude = c("filter", "lag")) # load it without the culprits
dlag    = dplyr::lag      # and fix ...
dfilter = dplyr::filter  # ... the blunders
```

then use `dlag` and `dfilter` for `dplyr` scripts and `lag` and `filter` can be use as originally intended.

- (3) Or just take back the commands:

```
filter = stats::filter
lag     = stats::lag
```

and you can still use these for `dplyr`:

```
dlag    = dplyr::lag
dfilter = dplyr::filter
```

- (4) Or avoid all of these problems and use `data.table` instead of `dplyr`. If you are doing data manipulation, you should know that `dplyr` is inspired by `data.table`, but it is much slower and weaker than `data.table`.

## Author(s)

David Stoffer <stoffer@pitt.edu>

## References

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

acf1

*Plot and print ACF or PACF of a time series***Description**

Produces a plot (and a printout) of the sample ACF or PACF. The zero lag value of the ACF is removed.

**Usage**

```
acf1(series, max.lag = NULL, plot = TRUE, main = NULL, ylim = NULL, pacf = FALSE,
      ylab = NULL, xlab = NULL, na.action = na.pass, ...)
```

**Arguments**

series	The data (must be more than 2 observations). Does not have to be a time series object.
max.lag	Maximum lag. Can be omitted. Unless $n < 60$ , defaults to $\sqrt{n} + 10$ or at least 4 seasons if the series is seasonal.
plot	If TRUE (default), a graph is produced and the values are rounded and listed. If FALSE, no graph is produced and the values are listed but not rounded by the script.
main	Title of graphic; defaults to name of series.
ylim	Specify limits for the y-axis.
pacf	If TRUE, the sample PACF is returned instead of ACF.
ylab	Change y-axis label from default.
xlab	Change x-axis label from default.
na.action	How to handle missing data; default is na.pass
...	Additional arguments passed to <a href="#">tsplot</a>

**Details**

Will print and/or plot the sample ACF or PACF (if pacf=TRUE). The zero lag of the ACF (which is always 1) has been removed. If plot=TRUE, a graph is produced and the values are rounded and listed. If plot=FALSE, no graph is produced and the values are listed but not rounded by the script. The error bounds are approximate white noise bounds,  $-1/n \pm 2/\sqrt{n}$ ; no other option is given.

**Value**

ACF	The sample ACF or PACF
-----	------------------------

**Author(s)**

D.S. Stoffer

## References

You can find demonstrations of astsa capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

## See Also

[acf2](#), [acfm](#), [ccf2](#)

## Examples

```
acf1(rnorm(100))
```

```
acf1(sarima.sim(ar=.9), pacf=TRUE)
```

```
acf1(soi, col=2:7, lwd=4, gg=TRUE)
```

---

acf2

*Plot and print ACF and PACF of a time series*

---

## Description

Produces a simultaneous plot (and a printout) of the sample ACF and PACF on the same scale. The zero lag value of the ACF is removed.

## Usage

```
acf2(series, max.lag = NULL, plot = TRUE, main = NULL, ylim = NULL,
      na.action = na.pass, ...)
```

## Arguments

<code>series</code>	The data (must be more than 2 observations). Does not have to be a time series object.
<code>max.lag</code>	Maximum lag. Can be omitted. Defaults to $\sqrt{n} + 10$ unless $n < 60$ . If the series is seasonal, this will be at least 4 seasons by default.
<code>plot</code>	If TRUE (default), a graph is produced and the values are rounded and listed. If FALSE, no graph is produced and the values are listed but not rounded by the script.
<code>main</code>	Title of graphic; defaults to name of series.
<code>ylim</code>	Specify limits for the y-axis.
<code>na.action</code>	How to handle missing data; default is <code>na.pass</code>
<code>...</code>	Additional arguments passed to <a href="#">tspplot</a>

**Details**

Will print and/or plot the sample ACF and PACF on the same scale. The zero lag of the ACF (which is always 1) has been removed. If `plot=TRUE`, a graph is produced and the values are rounded and listed. If `plot=FALSE`, no graph is produced and the values are listed but not rounded by the script. The error bounds are approximate white noise bounds,  $-1/n \pm 2/\sqrt{n}$ ; no other option is given.

**Value**

ACF	The sample ACF
PACF	The sample PACF

**Author(s)**

D.S. Stoffer

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**See Also**

[acf1](#), [acfm](#), [ccf2](#)

**Examples**

```
acf2(rnorm(100))

acf2(rnorm(100), 25, main=NA) # no title

acf2(rnorm(100), max.lag=5, plot=FALSE) # print only

acf2(soi, col=2:7, lwd=4, gg=TRUE)
```

---

acfm

*ACF and CCF for Multiple Time Series*

---

**Description**

Produces a grid of plots of the sample ACF (diagonal) and CCF (off-diagonal). The values are returned invisibly.

**Usage**

```
acfm(series, max.lag = NULL, na.action = na.pass, ylim = NULL,
      acf.highlight = TRUE, plot = TRUE, ...)
```

**Arguments**

<code>series</code>	Multiple time series (at least 2 columns of time series with more than 3 observations).
<code>max.lag</code>	Maximum lag. Can be omitted. Unless $n < 60$ , defaults to $\sqrt{n} + 10$ or at least 4 seasons if the series is seasonal.
<code>na.action</code>	How to handle missing data; default is <code>na.pass</code>
<code>ylim</code>	Specify limits for the all correlation axes. If <code>NULL</code> (default) the values are a little wider than the min and max of all values.
<code>acf.highlight</code>	If <code>TRUE</code> (default), the diagonals (ACFs) are highlighted.
<code>plot</code>	If <code>TRUE</code> (default), you get a wonderful graphic.
<code>...</code>	Additional arguments passed to <code>tsplot</code>

**Details**

Produces a grid of plots of the sample ACF (diagonal) and CCF (off-diagonal). The plots in the grid are estimates of  $\text{corr}\{x(t+\text{LAG}), y(t)\}$ . Thus  $x$  leads  $y$  if  $\text{LAG}$  is positive and  $x$  lags  $y$  if  $\text{LAG}$  is negative. If `plot` is `FALSE`, then there is no graphic.

**Value**

The correlations are returned invisibly.

**Author(s)**

D.S. Stoffer

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**See Also**

[acf1](#), [acf2](#), [ccf2](#)

**Examples**

```
acfm(diff(log(econ5)), gg=TRUE)

acfm(diff(log(econ5)), 2, plot=FALSE)
```

ar.boot

*Bootstrap Distribution of AR Model Parameters***Description**

Performs a nonparametric bootstrap to obtain the distribution of the AR model parameters.

**Usage**

```
ar.boot(series, order.ar, nboot = 500, seed = NULL, plot = TRUE, ...)
```

**Arguments**

series	time series data (univariate only)
order.ar	autoregression order - must be specified
nboot	number of bootstrap iterations (default is 500)
seed	seed for the bootstrap sampling (default is NULL)
plot	if TRUE (default) and order.ar > 1, returns a scatterplot matrix of the bootstrapped parameters, - the diagonals show a histogram.
...	if plot=TRUE and order.ar > 1, graphical parameters sent to <code>tspairs</code>

**Details**

For a specified series, finds the bootstrap distribution of the Yule-Walker estimates of  $\phi_1, \dots, \phi_p$  in the AR model specified by order.ar,

$$x_t = \mu + \phi_1(x_{t-1} - \mu) + \dots + \phi_p(x_{t-p} - \mu) + w_t,$$

where  $w_t$  is white noise. The data are centered by the sample mean prior to the bootstrap simulations.

The script displays a number of quantiles of the bootstrapped estimates, the means, the biases, and the root mean squared errors.

**Value**

Returned invisibly as a list:

phi.star	[[1]] bootstrapped AR parameters
x.sim	[[2]] bootstrapped data
mean.star	[[3]] bootstrapped mean
var.star	[[4]] bootstrapped noise variance
yw.fit	[[5]] results of Yule-Walker fit to the data

**Author(s)**

D.S. Stoffer

**References**

You can find demonstrations of astsa capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**Examples**

```
## Not run:

u = ar.boot(rec, 2)
dev.new()
tspairs(u[[1]], hist=FALSE) # another view of results
head(u[[1]]) # some booted AR parameters
head(u[[2]][,1:5]) # some booted data

## End(Not run)
```

---

ar.mcmc

*Fit Bayesian AR Model*


---

**Description**

Uses Gibbs sampling to fit an AR model to time series data.

**Usage**

```
ar.mcmc(xdata, porder, n.iter = 1000, n.warmup = 100, plot = TRUE,
        prior_var_phi = 50, prior_sig_a = 1, prior_sig_b = 2, ...)
```

**Arguments**

xdata	time series data (univariate only)
porder	autoregression order
n.iter	number of iterations for the sampler
n.warmup	number of startup iterations for the sampler (these are removed)
plot	if TRUE (default) returns the draws after warmup (diagonal) and a scatterplot matrix of the draws (off-diagonal)
prior_var_phi	prior variance of the vector of AR coefficients; see details

prior\_sig\_a first prior for the variance component; see details  
 prior\_sig\_b second prior for the variance component; see details  
 ... additional graphic parameters passed to tspairs

### Details

Assumes a normal-inverse gamma model,

$$x_t = \phi_0 + \phi_1 x_{t-1} + \dots + \phi_p x_{t-p} + \sigma z_t,$$

where  $z_t$  is standard Gaussian noise. With  $\Phi$  being the  $(p+1)$ -dimensional vector of the  $\phi$ s, the priors are  $\Phi \mid \sigma \sim N(0, \sigma^2 V_0)$  and  $\sigma^2 \sim IG(a, b)$ , where  $V_0 = \gamma^2 I$ . Defaults are given for the hyperparameters, but the user may choose  $(a, b)$  as `(prior_sig_a, prior_sig_b)` and  $\gamma^2$  as `prior_var_phi`.

The algorithm is efficient and converges quickly. Further details can be found in Chapter 6 of the 5th edition of the Springer text.

### Value

In addition to the graphics (if `plot` is TRUE), the draws of each parameter (`phi0`, `phi1`, ..., `sigma`) are returned invisibly and means, standard deviations, and various quantiles are displayed.

### Author(s)

D.S. Stoffer

### Source

Based on the script `arp.mcmc` used in Douc, Moulines, & Stoffer, D. (2014). *Nonlinear Time Series: Theory, Methods and Applications with R Examples*. CRC press. ISBN 9781466502253.

### References

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

### Examples

```
## Not run:

u = ar.mcmc(rec, 2)

tspairs(u, hist=FALSE, col.diag=6) # another view

tsplot(u, ncol=2, col=4, xlab='Index') # traces only
```

```
## End(Not run)
```

---

ar1miss	<i>AR with Missing Values</i>
---------	-------------------------------

---

### Description

Data used in Chapter 6 Problems

### Format

The format is: Time-Series [1:100] with NA for missing values.

### Details

Simulated AR(1) with  $\phi = .9$  and  $\sigma_w = 1$  and ~10% of the values missing at random (in this case, there happen to be 15 missing values). A similar data set can be generated as follows:

```
x = sarima.sim(ar=.9, n=100)
u = sample(c(NA,0), replace=TRUE, size=100, prob=c(.1,.9))
arm = x + u
```

Another approach if you want exactly 10 missing values is this:

```
arms = sarima.sim(ar=.9, n=100)
arms[sample(1:100, size=10)] = NA
```

### References

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

### Examples

```
# use type='o' when plotting a series with missing data
par(mfrow=2:1)
tsplot(ar1miss, col=4)
tsplot(ar1miss, col=4, type='o', pch=19)
```

arf

*Simulated ARFIMA***Description**

1000 simulated observations from an ARFIMA(1, 1, 0) model with  $\phi = .75$  and  $d = .4$ .

**Format**

The format is: Time-Series [1:1000] from 1 to 1000: -0.0294 0.7487 -0.3386 -1.0332 -0.2627 ...

**References**

You can find demonstrations of astsa capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**Examples**

```
par(mfrow=2:1)
tsplot(arf, col=4, main='ARFIMA')
acf1(arf, col=4, main=NA)
```

arma.check

*Check an ARMA Model for Causality, Invertibility, and Parameter Redundancy***Description**

For a given ARMA model (including seasonal models), reports whether the model is causal, invertible, or (approximately) over-parameterized.

**Usage**

```
arma.check(ar = 0, ma = 0, sar = NULL, sma = NULL, S = NULL, redtol = 0.1,
           plot.it = FALSE, ...)
```

**Arguments**

ar	vector of AR parameters
ma	vector of MA parameters
sar	vector of seasonal AR parameters (only specify for seasonal models)
sma	vector of seasonal MA parameters (only specify for seasonal models)
S	seasonal period (only specify for seasonal models - default value is 12)
redtol	tolerance for reporting parameter redundancy
plot.it	if TRUE -and- the model is causal and invertible, will plot the inverse roots and display the redundancy tolerance level, but ONLY for the AR and MA parts (seasonal parts are ignored)
...	additional graphical parameters

**Details**

Causality and invertibility are checked first. If either one or both are reported, checking is stopped. If the model is causal and invertible, a warning for (possible) over-parameterization/redundancy is given if there are (approximate) common zeros.

To evaluate parameter redundancy, the inverse roots of the AR and MA polynomials are examined for closeness with `redtol` determining closeness; see the note.

For fun, and IF the model is causal and invertible, setting `plot.it=TRUE` will display the complex plane with the inverse roots of the AR and MA polynomials displayed with colored arrows; the seasonal components are not included because it's too messy. The tolerance level for declaring over-parameterization is also displayed in the graphic.

**Value**

See the details. If the model is causal and invertible and not over-parameterized, a nice message of validation is given. Otherwise, problems are reported but the specific culprits may not be specified.

**Note**

The Fisher Information matrix for ARMA is singular for exact redundancy, and its closeness to singularity can be measured by the closeness of the inverse of those roots [e.g., Klein and Spreij (1997). On Fisher's information matrix of an ARMA process. In *Stochastic Differential and Difference Equations*, 273-284. Boston: Birkhauser.]

**Author(s)**

D.S. Stoffer

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

### See Also

[arma.spec](#), [sarima.sim](#), [ARMAtoAR](#), [ARMAtoMA](#)

### Examples

```
arma.check(ar=c(1,-.9), sar=-.6, sma=-.4, S=4)

arma.check(ar=.9, ma=c(-.9,-.8), sar=1, S=12)

# hard to tell from parameters alone ...
arma.check(ar=c(1.5,-.75), ma=c(-.6,-.3,.45), plot.it=TRUE, gg=TRUE)
```

---

arma.spec	<i>Spectral Density of an ARMA Model</i>
-----------	--

---

### Description

Gives the ARMA model spectrum, but first tests for causality and invertibility, and then for parameter redundancy.

### Usage

```
arma.spec(ar = 0, ma = 0, var.noise = 1, n.freq = 500, main = NULL, redundancy.tol=.1,
          frequency = 1, ylim = NULL, plot = TRUE, ...)
```

### Arguments

ar	vector of AR parameters
ma	vector of MA parameters
var.noise	variance of the noise
n.freq	number of frequencies
main	title of graphic; default is "ARMA" with orders "(p, q)"
redundancy.tol	tolerance for reporting parameter redundancy
frequency	for plotting, adjusts the frequency axis units
ylim	optional; specify limits for the y-axis
plot	if TRUE (default), produces a graphic
...	additional arguments for the graphic

## Details

The basic call is `arma.spec(ar, ma)` where `ar` and `ma` are vectors containing the model parameters. Use `log='y'` if you want the plot on a log scale.

If the model is not causal or invertible, a stop error with a message is given; e.g., `arma.spec(ar=1, ma=1)` will just produce warnings.

If there are (approximate) common zeros, a spectrum will be displayed and a warning will be given; e.g., `arma.spec(ar=.9, ma=-.9)` will yield a warning and the plot will be the spectrum of white noise. See [arma.check](#) for details on the evaluation of parameter redundancy.

## Value

<code>freq</code>	frequencies - returned invisibly
<code>spec</code>	spectral ordinates - returned invisibly

## Author(s)

D.S. Stoffer

## References

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

## See Also

[arma.check](#)

## Examples

```
arma.spec(ar = c(1, -.9), ma = .8)

arma.spec(ar = c(1, -.9), log='y')

arma.spec(ar = -.9, ma=.89, gg=TRUE) # almost white noise

# if you want a seasonal model, you have to be a little clever
arma.spec(ar=c(rep(0,11),.4), ma=.5, col=5, lwd=3, frequency=12)
```

ARMAtoAR

*Convert ARMA Process to Infinite AR Process***Description**

Gives the  $\pi$ -weights in the invertible representation of an ARMA model.

**Usage**

```
ARMAtoAR(ar = 0, ma = 0, lag.max=20)
```

**Arguments**

ar	vector of AR coefficients
ma	vector of MA coefficients
lag.max	number of pi-weights desired

**Value**

A vector of coefficients. For an ARMA model  $\phi(B)x_t = \theta(B)w_t$ , returns the coefficients  $\{\pi_j; j = 1, 2, \dots\}$  in the invertible representation  $w_t = \pi(B)x_t = \sum_{j=0}^{\infty} \pi_j x_{t-j}$  where  $\pi_0 = 1$ .

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**Examples**

```
ARMAtoAR(ar=.9, ma=.5, 10)
```

---

 astsa.col

*astsa color palette with transparency or a color wheel*


---

### Description

The script does one of two things. [1] Easily modify the opacity level of the astsa colors. [2] Create a color wheel of a chosen number of colors from a base color.

### Usage

```
astsa.col(col=1, alpha=1, wheel=FALSE, pie=FALSE, num, sat=NULL, val=NULL, ...)
```

### Arguments

col	Either <ol style="list-style-type: none"> <li>1. a numerical vector representing the colors to change transparency level -</li> <li>OR-</li> <li>2. a color used as a base for a wheel (if a vector, the first color is chosen)</li> </ol> - see Examples
alpha	factor in [0, 1] setting the opacity of all colors. Smaller values are more transparent.
wheel	if TRUE, produces a color wheel of num colors from a base color.
pie	if TRUE, produces a pie chart of the chosen colors.
num	an integer specifying the number of desired colors in the wheel. If missing, the user is prompted to enter a number.
sat	factor in [0, 1] setting the 'saturation' (intensity) of the colors. If NULL, the saturation from the base color is used.
val	factor in [0, 1] setting the 'value' (brightness) of the colors. If NULL, the value from the base color is used.
...	other graphical parameters for the pie chart; see <a href="#">pie</a> .

### Details

The astsa color palette is attached when the package is attached. The colors follow the R pattern of shades of: (1) black, (2) red, (3) green, (4) blue, (5) cyan, (6) magenta, (7) gold, (8) gray. The opacity of these colors can be changed easily using this script. Values are recycled, e.g., col=9 is the same as col=1.

Additionally, a color wheel can be generated by specifying a base color and inputting the number of desired colors. Using [hsv](#), the script moves the 'hue' around the circle in equal steps holding 'saturation' and 'value' constant. This may be overridden by entering an alternate 'saturation' or 'value'.

In either application, a pie chart can be displayed to help in choosing the desired color scheme.

**Value**

- [1] a *color vector* using the astsa color palette at the chosen transparency level
- OR -
- [2] a *color wheel* of a chosen number num of colors from a base color

**Author(s)**

D.S.Stoffer

**References**

You can find demonstrations of astsa capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**Examples**

```
# View the astsa palette
astsa.col(1:8, pie=TRUE)
legend('topright', legend=astsa.col(1:8), fill=1:8, title='Hex Color Code')

# Plotting 2 series that touch (but in a nice way)
tsplot(cbind(gtemp_land, gtemp_ocean), col=astsa.col(c(4,2), .5), lwd=2, spaghetti=TRUE,
        type='o', pch=20, ylab="Temperature Deviations", addLegend=TRUE, location='topleft',
        legend=c("Land Only", "Ocean Only"), gg=TRUE)

# The hsv values for Dodgerblue3 (or astsa color 4)?
rgb2hsv(col2rgb(4))

# Wheels of fortune
vanna = par(no.readonly = TRUE)
par(mar=rep(0, 4))
layout( matrix(c(1,3,2,3), 2) )
astsa.col(4, wheel=TRUE, num=8, pie=TRUE)
astsa.col(4, wheel=TRUE, num=8, pie=TRUE, sat=.6, val=.9)
astsa.col(2, wheel=TRUE, num=100, pie=TRUE, border=FALSE, labels=NA)

# I'd like to solve the puzzle
par(vanna) # reset graphic device
x = replicate(6, sarima.sim(ar=c(1.5,-.75), n=120))
tsplot(x, spag=TRUE, col=astsa.col(4, alpha=.7, wheel=TRUE, num=6), lwd=12)
```

**Description**

Uses minimum description length (MDL) to fit piecewise AR processes with the goal of detecting changepoints in time series. Optimization is accomplished via a genetic algorithm (GA).

**Usage**

```
autoParm(xdata, Pi.B = NULL, Pi.C = NULL, PopSize = 70, generation = 70, P0 = 20,
        Pi.P = 0.3, Pi.N = 0.3, NI = 7)
```

**Arguments**

xdata	time series (of length n at least 100) to be analyzed; the ts attributes are stripped prior to the analysis
Pi.B	probability of being a breakpoint in initial stage; default is 10/n. Does not need to be specified.
Pi.C	probability of conducting crossover; default is (n-10)/n. Does not need to be specified.
PopSize	population size (default is 70); the number of chromosomes in each generation. Does not need to be specified.
generation	number of iterations; default is 70. Does not need to be specified.
P0	maximum AR order; default is 20. If larger than 20, it is reset to 20. Does not need to be specified.
Pi.P	probability of taking parent's gene in mutation; default is 0.3. Does not need to be specified.
Pi.N	probability of taking -1 in mutation; default is 0.3 Does not need to be specified.
NI	number if islands; default is 7. Does not need to be specified.

**Details**

Details may be found in Davis, Lee, & Rodriguez-Yam (2006). Structural break estimation for nonstationary time series models. JASA, 101, 223-239. [doi:10.1198/016214505000000745](https://doi.org/10.1198/016214505000000745)

**Value**

Returns three values, (1) the breakpoints including the endpoints, (2) the number of segments, and (3) the segment AR orders. See the examples.

**Note**

The GA is a stochastic optimization procedure and consequently will give different results at each run. It is a good idea to run the algorithm a few times before coming to a final decision.

**Author(s)**

D.S. Stoffer

**Source**

The code is adapted from R code provided to us by Rex Cheung (<https://www.linkedin.com/in/rexcheung>).

**References**

You can find demonstrations of astsa capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**See Also**

[autoSpec](#)

**Examples**

```
## Not run:

##-- simulation
x1 = sarima.sim(ar=c(1.69, -.81), n=500)
x2 = sarima.sim(ar=c(1.32, -.81), n=500)
x = c(x1, x2)

##-- look at the data
tsplot(x)

##-- run procedure
autoParm(x)

##-- output (yours will be slightly different -
##--         the nature of GA)
# returned breakpoints include the endpoints
# $breakpoints
# [1] 1 514 1000
#
# $number_of_segments
# [1] 2
#
# $segment_AR_orders
# [1] 2 2

## End(Not run)
```

autoSpec

*autoSpec - Changepoint Detection of Narrowband Frequency Changes***Description**

Uses changepoint detection to discover if there have been slight changes in frequency in a time series. The autoSpec procedure uses minimum description length (MDL) to do nonparametric spectral estimation with the goal of detecting changepoints. Optimization is accomplished via a genetic algorithm (GA).

**Usage**

```
autoSpec(xdata, Pi.B = NULL, Pi.C = NULL, PopSize = 70, generation = 70, m0 = 10,
        Pi.P = 0.3, Pi.N = 0.3, NI = 7, taper = .5, min.freq = 0, max.freq = .5)
```

**Arguments**

xdata	time series (of length n at least 100) to be analyzed; the ts attributes are stripped prior to the analysis
Pi.B	probability of being a breakpoint in initial stage; default is 10/n. Does not need to be specified.
Pi.C	probability of conducting crossover; default is (n-10)/n. Does not need to be specified.
PopSize	population size (default is 70); the number of chromosomes in each generation. Does not need to be specified.
generation	number of iterations; default is 70. Does not need to be specified.
m0	maximum width of the Bartlett kernel is 2*m0 + 1; default is 10. If larger than 20, m0 is reset to 20. Does not need to be specified.
Pi.P	probability of taking parent's gene in mutation; default is 0.3. Does not need to be specified.
Pi.N	probability of taking -1 in mutation; default is 0.3 Does not need to be specified.
NI	number if islands; default is 7. Does not need to be specified.
taper	half width of taper used in spectral estimate; .5 (default) is full taper. Does not need to be specified.
min.freq, max.freq	the frequency range (min.freq, max.freq) over which to calculate the Whittle likelihood; the default is (0, .5). Does not need to be specified. If min > max, the roles are reversed, and reset to the default if either is out of range.

**Details**

Details may be found in Stoffer, D. S. (2023). AutoSpec: Detection of narrowband frequency changes in time series. *Statistics and Its Interface*, 16(1), 97-108. [doi:10.4310/21SI1703](https://doi.org/10.4310/21SI1703)

**Value**

Returns three values, (1) the breakpoints including the endpoints, (2) the number of segments, and (3) the segment kernel orders. See the examples.

**Note**

The GA is a stochastic optimization procedure and consequently will give different results at each run. It is a good idea to run the algorithm a few times before coming to a final decision.

**Author(s)**

D.S. Stoffer

**Source**

The genetic algorithm code is adapted from R code provided to us by Rex Cheung (<https://www.linkedin.com/in/rexcheung/>). The code originally supported Aue, Cheung, Lee, & Zhong (2014). Segmented model selection in quantile regression using the minimum description length principle. *JASA*, 109, 1241-1256. A similar version also supported Davis, Lee, & Rodriguez-Yam (2006). Structural break estimation for nonstationary time series models. *JASA*, 101, 223-239.

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**See Also**

[autoParm](#)

**Examples**

```
## Not run:

##-- simulation
set.seed(1)
num = 500
t = 1:num
w = 2*pi/25
d = 2*pi/150
x1 = 2*cos(w*t)*cos(d*t) + rnorm(num)
x2 = cos(w*t) + rnorm(num)
x = c(x1,x2)

##-- plot and periodogram (all action below 0.1)
```

```

tsplot(x, main='not easy to see the change')
mvspec(x)

##-- run procedure
autoSpec(x, max.freq=.1)

##-- output (yours will be slightly different -
##--         the nature of GA)
# returned breakpoints include the endpoints
# $breakpoints
# [1] 1 503 1000
#
# $number_of_segments
# [1] 2
#
# $segment_kernel_orders_m
# [1] 2 4

##-- plot everything
par(mfrow=c(3,1))
tsplot(x, col=4)
abline(v=503, col=6, lty=2, lwd=2)
mvspec(x[1:502], kernel=bart(2), taper=.5, main='segment 1', col=4, xlim=c(0,.25))
mvspec(x[503:1000], kernel=bart(4), taper=.5, main='segment 2', col=4, xlim=c(0,.25))

## End(Not run)

```

---

bart

*Bartlett Kernel*


---

### Description

Smoothing (triangular) kernel that decreases one unit from the center.

### Usage

```
bart(m)
```

### Arguments

**m** non-negative integer specifying the kernel width, which is  $2m + 1$ . If **m** has length larger than one, the convolution of the kernel is returned.

### Details

Uses kernel from the `stats` package to construct a Bartlett (triangular) kernel of width  $2m + 1$ ; see `help(kernel)` for further details.

**Value**

Returns an object of class `tskernel` with the coefficients, the kernel dimension, and attribute "Bartlett".

**Author(s)**

D.S. Stoffer

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**Examples**

```
bart(4) # for a list
plot(bart(4), ylim=c(.01,.21)) # for a graph
```

---

BCJ

*Daily Returns of Three Banks*

---

**Description**

Daily returns of three banks, 1. Bank of America [boa], 2. Citibank [citi], and 3. JP Morgan Chase [jpm], from 2005 to 2017.

**Format**

The format is: Time-Series [1:3243, 1:3] from 2005 to 2017: -0.01378 -0.01157 -0.00155 -0.01084 0.01252 ... with column names "boa" "citi" "jpm" .

**Source**

Gong & Stoffer (2021). A Note on Efficient Fitting of Stochastic Volatility Models. *Journal of Time Series Analysis*, 42(2), 186-200.

<https://github.com/nickpoison/Stochastic-Volatility-Models>

## References

You can find demonstrations of astsa capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

## Examples

```
tsplot(BCJ, col=2:4)
```

---

beamd

*Infrasonic Signal from a Nuclear Explosion*

---

## Description

Infrasonic signal from a nuclear explosion.

## Usage

```
data(beamd)
```

## Format

A data frame with 2048 observations (rows) on 3 numeric variables (columns): sensor1, sensor2, sensor3.

## Details

This is a data frame consisting of three columns (that are not time series objects). The data are an infrasonic signal from a nuclear explosion observed at sensors on a triangular array.

## References

You can find demonstrations of astsa capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

---

birth *U.S. Monthly Live Births*

---

### Description

Monthly live births (adjusted) in thousands for the United States, 1948-1979.

### Format

The format is: Time-Series [1:373] from 1948 to 1979: 295 286 300 278 272 268 308 321 313 308 ...

### References

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

---

blood *Daily Blood Work with Missing Values*

---

### Description

Multiple time series of measurements made for 91 days on the three variables, log(white blood count) [WBC], log(platelet) [PLT] and hematocrit [HCT]. Missing data code is NA.

### Format

Time-Series [1:91, 1:3] from 1 to 91: 2.33 1.89 2.08 1.82 1.82 ...  
 ..\$ : NULL ..\$ : chr [1:3] "WBC" "PLT" "HCT"

### Details

This data set is used in Chapter 6 for a missing data example.

### Source

Jones, R.H. (1984). Fitting multivariate models to unequally spaced data. In *Time Series Analysis of Irregularly Observed Data*, pp. 158-188. E. Parzen, ed. Lecture Notes in Statistics, 25, New York: Springer-Verlag.

## References

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

## See Also

[HCT](#), [PLT](#), [WBC](#)

## Examples

```
# use type='o' when plotting series with missing data
tsplot(blood, type='o', pch=19, cex=1.1, col=2:4, gg=TRUE, xlab='day')
```

---

bnrf1ebv

*Nucleotide sequence - B NRF1 Epstein-Barr*

---

## Description

Nucleotide sequence of the B NRF1 gene of the Epstein-Barr virus (EBV): 1=A, 2=C, 3=G, 4=T. The data are used in Chapter 7.

## Format

The format is: Time-Series [1:3954] from 1 to 3954: 1 4 3 3 1 1 3 1 3 1 ...

## References

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

## Examples

```
head(bnrf1ebv)
head(dna2vector(bnrf1ebv))
```

---

bnrf1hvs

*Nucleotide sequence - BNRF1 of Herpesvirus saimiri*

---

### Description

Nucleotide sequence of the BNRF1 gene of the herpesvirus saimiri (HVS): 1=A, 2=C, 3=G, 4=T. The data are used in Chapter 7.

### Format

The format is: Time-Series [1:3741] from 1 to 3741: 1 4 3 2 4 4 3 4 4 4 ...

### References

You can find demonstrations of astsa capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

### Examples

```
head(bnrf1hvs)
head(dna2vector(bnrf1hvs))
```

---

cardox

*Monthly Carbon Dioxide Levels at Mauna Loa*

---

### Description

Monthly mean carbon dioxide (in ppm) measured at Mauna Loa Observatory, Hawaii. This is an update to co2 in the datasets package.

### Format

The format is: Time-Series [1:781] from 1958 to 2023: 316 317 318 317 316 ...

## Details

The carbon dioxide data measured as the mole fraction in dry air, on Mauna Loa constitute the longest record of direct measurements of CO<sub>2</sub> in the atmosphere. They were started by C. David Keeling of the Scripps Institution of Oceanography in March of 1958 at a facility of the National Oceanic and Atmospheric Administration. NOAA started its own CO<sub>2</sub> measurements in May of 1974, and they have run in parallel with those made by Scripps since then. Data are reported as a dry mole fraction defined as the number of molecules of carbon dioxide divided by the number of molecules of dry air multiplied by one million (ppm).

Due to the eruption of the Mauna Loa Volcano, measurements from Mauna Loa Observatory were suspended as of Nov. 29, 2022. Observations starting in December 2022 are from a site at the Maunakea Observatories, approximately 21 miles north of the Mauna Loa Observatory.

## Note

The reason this data file is called `cardox` is because R has two datasets called `CO2` and `co2`, which doesn't leave many options if you don't want to create potential problems like the `dplyr` people. Anyway, `cardox` sounds like the name of a cool character in some SciFi movie.

## Source

<https://gml.noaa.gov/ccgg/trends/>

## References

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

---

ccf2

*Cross Correlation*

---

## Description

Calculates and plots the sample CCF of two time series.

## Usage

```
ccf2(x, y, max.lag = NULL, main = NULL, ylab = "CCF", plot = TRUE,  
      na.action = na.pass, type = c("correlation", "covariance"), ...)
```

**Arguments**

<code>x, y</code>	univariate time series
<code>max.lag</code>	maximum lag for which to calculate the CCF. Can be omitted. Unless $n < 50$ , defaults to $\sqrt{n} + 10$ or at least 3 seasons if the series is seasonal.
<code>main</code>	plot title - if NULL, uses x and y names
<code>ylab</code>	vertical axis label; default is 'CCF'
<code>plot</code>	if TRUE (default) a graphic is produced and the values are returned invisibly. Otherwise, the values are returned.
<code>na.action</code>	how to handle missing values; default is <code>na.pass</code>
<code>type</code>	default is cross-correlation; an option is cross-covariance
<code>...</code>	additional arguments passed to <code>tsplot</code>

**Details**

This will produce a graphic of the sample  $\text{corr}[x(t+\text{lag}), y(t)]$  from  $-\text{max.lag}$  to  $\text{max.lag}$ . Also, the (rounded) values of the CCF are returned invisibly unless `plot=FALSE`. Similar details apply to the cross-covariance.

**Author(s)**

D.S. Stoffer

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**See Also**

[pre.white](#), [acf1](#), [acf2](#), [acfm](#)

**Examples**

```
ccf2(soi, rec, plot=FALSE, max.lag=6) # now you see it
ccf2(soi, rec)                        # now you don't

# happy birthday
ccf2(soi, rec, col=astsa.col(4, wheel=TRUE, num=6), lwd=4, gg=TRUE)
```

---

chicken	<i>Monthly price of a pound of chicken</i>
---------	--

---

**Description**

Poultry (chicken), Whole bird spot price, Georgia docks, US cents per pound

**Format**

The format is: Time-Series [1:180] from August 2001 to July 2016: 65.6 66.5 65.7 64.3 63.2 ...

**Source**

<https://www.indexmundi.com/commodities/>

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

---

climhyd	<i>Lake Shasta inflow data</i>
---------	--------------------------------

---

**Description**

Lake Shasta inflow data. This is a data frame.

**Format**

A data frame with 454 observations (rows) on the following 6 numeric variables (columns): Temp, DewPt, CldCvr, WndSpd, Precip, Inflow.

**Details**

The data are 454 months of measured values for the climatic variables: air temperature, dew point, cloud cover, wind speed, precipitation, and inflow, at Lake Shasta, California.

Lake Shasta is a man-made reservoir that is situated on ancestral lands of the Shasta Indian Nation. The area around the lake was once a thriving homeland for the Shasta. The construction of the reservoir submerged many of these significant cultural sites, leading to displacement and hardship for the Shasta people. The California Gold Rush drew in outsiders in the late 1840s. For the Shasta, this was a devastating process as thousands of miners who did not respect the Shasta or their homeland, operated along various waterways. Introduction of diseases and fighting against white people, rapidly reduced the number of Shasta. For more information, see <https://www.legendsofamerica.com/shasta-indians>.

## References

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

## Examples

```
tsplot(climhyd, ncolm=2, col=2:7, lwd=2, scale=.9)
```

---

cmort

*Cardiovascular Mortality from the LA Pollution study*

---

## Description

Average weekly cardiovascular mortality in Los Angeles County; 508 six-day smoothed averages obtained by filtering daily values over the 10 year period 1970-1979.

## Format

The format is: Time-Series [1:508] from 1970 to 1980: 97.8 104.6 94.4 98 95.8 ...

## References

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

## See Also

[lap](#), [lap.xts](#)

---

cpg	<i>Hard Drive Cost per GB</i>
-----	-------------------------------

---

**Description**

Median annual cost per gigabyte (GB) of storage.

**Format**

The format is: Time-Series [1:29] from 1980 to 2008: 213000.00 295000.00 260000.00 175000.00 160000.00 ...

**Note**

This is a good data set for a first exercise on regression with autocorrelated errors.

**Source**

The data are retail prices per GB taken from a sample of manufacturers listed at <https://mkomo.com/cost-per-gigabyte>. From there, the median cost per GB was computed by year.

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

---

detrend	<i>Detrend a Time Series</i>
---------	------------------------------

---

**Description**

Returns a time series with the trend removed. The trend can be estimated using polynomial regression or using a lowess fit.

**Usage**

```
detrend(series, order = 1, lowess = FALSE, lowspan = 2/3)
```

**Arguments**

series	The time series to be detrended.
order	Order of the polynomial used to estimate the trend with a linear default (order=1) unless lowess is TRUE.
lowess	If TRUE, lowess is used to find the trend. The default is FALSE.
lowspan	The smoother span used for lowess.

**Value**

The detrended series is returned.

**Author(s)**

D.S. Stoffer

**References**

You can find demonstrations of astsa capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**See Also**

[trend](#)

**Examples**

```
tsplot( cbind(salmon, detrend(salmon)), gg=TRUE, main='Norwegian Salmon USD/KG' )
```

---

djia

*Dow Jones Industrial Average*

---

**Description**

Daily DJIA values from April 2006 - April 2016

**Format**

The format is: xts [1:2518, 1:5] 11279 11343 11347 11337 11283 ...  
 - attr(\*, "class")= chr [1:2] "xts" "zoo"  
 ..\$ : chr [1:5] "Open" "High" "Low" "Close" "Volume"

**Source**

The data were obtained via the TTR package and Yahoo financial data. Unfortunately, this does not work now. It seems like the R package `quantmod` is a good bet and Yahoo still has financial data.

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**Examples**

```
# the file is an 'xts' data file
# if 'xts' is not installed you can still do this

tsplot(djia, ncolm=2, col=2:6)           # no dates
tsplot(timex(djia), djia, ncolm=2, col=2:6) # dates

# differencing (on its own) loses an obs
return = diff(log(djia[, 'Close']))
tsplot(timex(djia)[-1], return, col=4, gg=TRUE, main='DJIA')
```

---

dna2vector

*Convert DNA Sequence to Indicator Vectors*

---

**Description**

Takes a string (e.g., a DNA sequence) of general form (e.g., FASTA) and converts it to a sequence of indicator vectors for use with the Spectral Envelope (`specenv`).

**Usage**

```
dna2vector(data, alphabet = NULL)
```

**Arguments**

<code>data</code>	A single string.
<code>alphabet</code>	The particular alphabet being used. The default is <code>alphabet=c("A", "C", "G", "T")</code> .

## Details

Takes a string of categories and converts it to a matrix of indicators. The data can then be used by the script [specenv](#), which calculates the Spectral Envelope of the sequence (or subsequence). Many different type of sequences can be used, including FASTA and GenBank, as long as the data is a string of categories.

The indicator vectors (as a matrix) are returned invisibly in case the user forgets to put the results in an object wherein the screen would scroll displaying the entire sequence. In other words, the user should do something like `xdata = dna2vector(data)` where `data` is the original sequence.

If the DNA sequence is in a FASTA file, say `sequence.fasta`, the following code can be used to read the data into the session, create the indicator sequence and save it as a compressed R data file:

```
fileName <- 'sequence.fasta'      # name of FASTA file
data      <- readChar(fileName, file.info(fileName)$size) # input the sequence
myseq    <- dna2vector(data)      # convert it to indicators

##== to compress and save the data ==##
save(myseq, file='myseq.rda')
##== and then load it when needed ==##
load('myseq.rda')
```

## Value

matrix of indicator vectors; returned invisibly

## Author(s)

D.S. Stoffer

## References

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

## See Also

[specenv](#)

## Examples

```
# Epstein-Barr virus (entire sequence included in astsa)
substr(EBV, start=1, stop=25)
xdata = dna2vector(EBV)
head(xdata)
```

```

# part of EBV with 1, 2, 3, 4 for "A", "C", "G", "T"
xdata = dna2vector(bnrf1ebv)
head(xdata)

# raw GenBank sequence
data <-
c("1 agaattcgtc ttgctctatt cacccttact tttcttcttg cccgttctct ttcttagtat
  61 gaatccagta tgcctgcctg taattgttg gcctacctc ttttgctgg cggctattgc")
xdata = dna2vector(data, alphabet=c('a', 'c', 'g', 't'))
head(xdata)

# raw FASTA sequence
data <-
c("AGAATTCGTCTTGCTCTATTCACCCTTACTTTTCTTCTTGCCGTTCTCTTCTTAGTATGAATCCAGTA
  TGCTGCCTGTAATTGTTGCGCCCTACCTCTTTGGCTGGCGGCTATTGCCGCTCGTGTTCACGGCCT")
xdata = dna2vector(data)
head(xdata)

```

---

EBV

*Entire Epstein-Barr Virus (EBV) Nucleotide Sequence*


---

## Description

EBV nucleotide sequence - 172281 bp as a single string

## Format

The format is: chr "AGAATTCGTCTT ..."

## Note

EBV is not useful on its own, but using 'dna2vector', different regions can be explored. For example, `ebv = dna2vector(EBV)`

## Source

<https://www.ncbi.nlm.nih.gov/nuccore/V01555.2>

## References

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**See Also**[dna2vector](#)**Examples**

```
substr(EBV, start=1, stop=25)
head(dna2vector(EBV))
```

---

econ5

*Five Quarterly Economic Series*

---

**Description**

Multiple time series of quarterly U.S. unemployment, GNP, consumption, and government and private investment, from 1948-III to 1988-II.

**Format**

Multiple time series with 161 observations (rows) on the following 5 numeric variables (columns): unemp, gnp, consum, govinv, prinv.

**Note**

unemp is in percentage (of the labor force), and the others are in 1982 Billions USD.

**Source**

Young, P.C. and Pedregal, D.J. (1999). Macro-economic relativity: government spending, private investment and unemployment in the USA 1948-1998. *Structural Change and Economic Dynamics*, 10, 359-380.

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**Examples**

```
tsplot(econ5, ncolm=2, col=2:6, lwd=2, title=colnames(econ5),
       ylab=c('%', rep('Billions-USD', 4)))
```

**Description**

Estimation of the parameters in general linear state space models via the EM algorithm. Missing data may be entered as NA or as zero (0), however, use NAs if zero (0) can be an observation. Inputs in both the state and observation equations are allowed. This script replaces EM0 and EM1.

**Usage**

```
EM(y, A, mu0, Sigma0, Phi, Q, R, Ups = NULL, Gam = NULL, input = NULL,
    max.iter = 100, tol = 1e-04)
```

**Arguments**

y	data matrix (n x q), vector or time series, n = number of observations, q = number of series. Use NA or zero (0) for missing data, however, use NAs if zero (0) can be an observation.
A	measurement matrices; can be constant or an array with dimension dim=c(q, p, n) if time varying. Use NA or zero (0) for missing data.
mu0	initial state mean vector (p x 1)
Sigma0	initial state covariance matrix (p x p)
Phi	state transition matrix (p x p)
Q	state error matrix (p x p)
R	observation error matrix (q x q - diagonal only)
Ups	state input matrix (p x r); leave as NULL (default) if not needed
Gam	observation input matrix (q x r); leave as NULL (default) if not needed
input	NULL (default) if not needed or a matrix (n x r) of inputs having the same row dimension (n) as y
max.iter	maximum number of iterations
tol	relative tolerance for determining convergence

**Details**

This script replaces EM0 and EM1 by combining all cases and allowing inputs in the state and observation equations. It uses version 1 of the new [Ksmooth](#) script (hence correlated errors is not allowed).

The states  $x_t$  are p-dimensional, the data  $y_t$  are q-dimensional, and the inputs  $u_t$  are r-dimensional for  $t = 1, \dots, n$ . The initial state is  $x_0 \sim N(\mu_0, \Sigma_0)$ .

The general model is

$$\begin{aligned}x_t &= \Phi x_{t-1} + \Upsilon u_t + w_t & w_t &\sim iid N(0, Q) \\y_t &= A_t x_{t-1} + \Gamma u_t + v_t & v_t &\sim iid N(0, R)\end{aligned}$$

where  $w_t \perp v_t$ . The observation noise covariance matrix is assumed to be diagonal and it is forced to diagonal otherwise.

The measurement matrices  $A_t$  can be constant or time varying. If time varying, they should be entered as an array of dimension  $\text{dim} = c(q, p, n)$ . Otherwise, just enter the constant value making sure it has the appropriate  $q \times p$  dimension.

### Value

Phi	Estimate of Phi
Q	Estimate of Q
R	Estimate of R
Ups	Estimate of Upsilon (NULL if not used)
Gam	Estimate of Gamma (NULL if not used)
mu0	Estimate of initial state mean
Sigma0	Estimate of initial state covariance matrix
like	-log likelihood at each iteration
niter	number of iterations to convergence
cvg	relative tolerance at convergence

### Note

The script does not allow for constrained estimation directly, however, constrained estimation is possible with some extra manipulations. There is an example of constrained estimation using EM at [FUN WITH ASTSA](#), where the fun never stops.

### Author(s)

D.S. Stoffer

### References

You can find demonstrations of astsa capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

### See Also

[Kfilter](#), [Ksmooth](#)

**Examples**

```

y = ar1miss # an 'astsa' data set with missing values
A = 1; Phi = .1; Ups = 1
Q = 0.1; R = 0.1; Gam=1
mu0 = 0; Sigma0 = 1
input = rep(1, length(y))
( em = EM(y, A, mu0, Sigma0, Phi, Q, R, Ups, Gam, input) )

# run Kalman smoother at the final estimates
ks = Ksmooth(y, A=1, em$mu0, em$Sigma0, em$Phi, sQ=sqrt(em$Q), sR=sqrt(em$R),
             Ups=em$Ups, Gam=em$Gam, input=input)

# admire your work
tsplot(cbind(y, drop(ks$Xs)), type='o', pch=c(19,NA), col=2*2:3, lwd=1:2,
        gg=TRUE, spag=TRUE, addLegend=TRUE, legend=c('data', 'smooth'))
miss = cbind(which(is.na(y)), min(y, na.rm=TRUE)-.2) # show yourself
points(miss, pch=18)

```

---

 ENSO

*El Nino - Southern Oscillation Index*


---

**Description**

Southern Oscillation Index (SOI), 1/1951 to 10/2022; anomalies are departures from the 1981-2010 base period.

**Format**

The format is: Time-Series [1:862] from 1951 to 2022: 2.0 1.1 -0.3 -0.8 -1.1 -0.7 -1.5 -0.3 -0.7 -0.7 ...

**Details**

The El Niño - Southern Oscillation (ENSO) is a recurring climate pattern involving changes in the temperature of waters in the central and eastern tropical Pacific Ocean. This data set is an update to [soi](#).

**Source**

<https://www.ncei.noaa.gov/access/monitoring/enso/soi>

**References**

You can find demonstrations of astsa capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**See Also**

[soi](#), [ENSO](#), [MEI](#), [MEI2](#)

**Examples**

```
# ENSO is a standardized index based on the sea level pressure differences between
# Tahiti and Darwin, Australia. MEI is a similar index based on multiple factors.
# As opposed to ENSO (and SOI), positive values of MEI correspond to warmer temperatures.

tspplot(cbind(-ENSO, MEI), col=astsa.col(2*2:1, .6), addLegend=TRUE, spag=TRUE, lwd=1:2)

tspairs(ts.intersect(-ENSO, MEI), location='top')
```

EQ5

*Seismic Trace of Earthquake number 5***Description**

Seismic trace of an earthquake [two phases or arrivals along the surface, the primary wave ( $t = 1, \dots, 1024$ ) and the shear wave ( $t = 1025, \dots, 2048$ )] recorded at a seismic station.

**Format**

The format is: Time-Series [1:2048] from 1 to 2048: 0.01749 0.01139 0.01512 0.01477 0.00651 ...

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**See Also**

[eqexp](#)

---

EQcount

*Earthquake Counts*

---

### Description

Series of annual counts of major earthquakes (magnitude 7 and above) in the world between 1900 and 2006.

### Format

The format is: Time-Series [1:107] from 1900 to 2006: 13 14 8 10 16 26 ...

### Source

Zucchini and MacDonald (2009). Hidden Markov Models for Time Series: An Introduction using R. CRC Press.

### References

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

---

eqexp

*Earthquake and Explosion Seismic Series*

---

### Description

This is a data frame of the earthquake and explosion seismic series used throughout the text.

### Format

A data frame with 2048 observations (rows) on 17 variables (columns). Each column is a numeric vector.

## Details

The matrix has 17 columns, the first eight are earthquakes, the second eight are explosions, and the last column is the Novaya Zemlya event of unknown origin.

The column names are: EQ1, EQ2, . . . ,EQ8; EX1, EX2, . . . ,EX8; NZ. The first 1024 observations correspond to the P wave, the second 1024 observations correspond to the S wave.

All events in the data set were on or near land and were distributed uniformly over Scandinavia so as to minimize the possibility that discriminators might be keying on location or land-sea differences. The events are earthquakes ranging in magnitude from 2.74 to 4.40 and explosions in the range 2.13 to 2.19. Also added is an event of uncertain origin that was located in the Novaya Zemlya region of Russia. All events except the Russian event occurred in the Scandinavian peninsula and were recorded by seismic arrays located in Norway by Norwegian and Arctic experimental seismic stations (NORESS, ARCESS) and in Finland by Finnish experimental seismic stations (FINESS).

No.	Type	Date	Array	Magnitude	Latitude	Longitude
1	EQ	6/16/92	FINESS	3.22	65.5	22.9
2	EQ	8/24/91	ARCESS	3.18	65.7	32.1
3	EQ	9/23/91	NORESS	3.15	64.5	21.3
4	EQ	7/4/92	FINESS	3.60	67.8	15.1
5	EQ	2/19/92	ARCESS	3.26	59.2	10.9
6	EQ	4/13/92	NORESS	4.40	51.4	6.1
7	EQ	4/14/92	NORESS	3.38	59.5	5.9
8	EQ	5/18/92	NORESS	2.74	66.9	13.7
9	EX	3/23/91	ARCESS	2.85	69.2	34.3
10	EX	4/13/91	FINESS	2.60	61.8	30.7
11	EX	4/26/91	ARCESS	2.95	67.6	33.9
12	EX	8/3/91	ARCESS	2.13	67.6	30.6
13	EX	9/5/91	ARCESS	2.32	67.1	21.0
14	EX	12/10/91	FINESS	2.59	59.5	24.1
15	EX	12/29/91	ARCESS	2.96	69.4	30.8
16	EX	3/25/92	NORESS	2.94	64.7	30.8
17	NZ	12/31/92	NORESS	2.50	73.6	55.2

## References

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

## Examples

```
# view all series
# first 2 rows EQs - next 2 rows EXs
# 5th row NZ event
```

```

tsplot(eqexp, ncol=4, col=c(rep(6,8),rep(4,8),2), scale=.75)

## Not run:

# for a nice html map, use the 'leaflet' package

library(leaflet) # install if you don't have it

lat = c(
65.5,65.7,64.5,67.8,59.2,51.4,59.5,66.9,
69.2,61.8,67.6,67.6,67.1,59.5,69.4,64.7,
73.6)

long = c(
22.9,32.1,21.3,15.1,10.9,6.1,5.9,13.7,
34.3,30.7,33.9,30.6,21.0,24.1,30.8,30.8,
55.2)

event = c(
"EQ1","EQ2","EQ3","EQ4","EQ5","EQ6","EQ7","EQ8",
"EX1","EX2","EX3","EX4","EX5","EX6","EX7","EX8",
"NZ")

eventmap = data.frame(
  longitude = long,
  latitude = lat,
  event = event )

leaflet(eventmap) %>%
  addTiles() %>%
  addMarkers(~longitude, ~latitude, label = ~event,
    labelOptions = labelOptions(noHide = TRUE))

## End(Not run)

```

---

ESS

*Effective Sample Size (ESS)*


---

## Description

Estimates the ESS of a given vector of samples.

## Usage

```
ESS(trace, tol = 1e-08, BIC = TRUE, digits=2)
```

**Arguments**

<code>trace</code>	vector of sampled values from an MCMC run (univariate only)
<code>tol</code>	ESS is returned as zero if the estimated spectrum at frequency zero is less than this value
<code>BIC</code>	if TRUE (default), <code>spec0</code> is obtained using BIC; otherwise, AIC is used. See the details.
<code>digits</code>	integer indicating the approximate number of decimal places to be used

**Details**

Uses `spec.ic` to estimate the spectrum of the input at frequency zero (`spec0`). Then, ESS is estimated as  $ESS = \text{length}(\text{trace}) * \text{var}(\text{trace}) / \text{spec0}$ . See Examples for multivariate case.

ESS is discussed in detail in Example 6.31 of *Time Series Analysis and Its Applications: With R Examples* (5th ed, 2025).

**Value**

Returns the estimated ESS of the input.

**Author(s)**

D.S. Stoffer

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**Examples**

```
# How autocorrelation affects ESS
# get a few Markov chains
x = matrix(NA, 500, 3) # sample size is 500
phi = c(0,.3,.9) # no, low, high
for (i in 1:3) x[,i] = sarima.sim(ar=phi[i])
apply(x, 2, ESS, digits=0)
```

---

EXP6

*Seismic Trace of Explosion number 6*

---

### Description

Seismic trace of an explosion [two phases or arrivals along the surface, the primary wave ( $t = 1, \dots, 1024$ ) and the shear wave ( $t = 1025, \dots, 2048$ )] recorded at a seismic station.

### Format

The format is: Time-Series [1:2048] from 1 to 2048: -0.001837 -0.000554 -0.002284 -0.000303 -0.000721 ...

### References

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

### See Also

[eqexp](#)

---

FDR

*Basic False Discovery Rate*

---

### Description

Computes the basic false discovery rate given a vector of p-values and returns the index of the maximal p-value satisfying the FDR condition.

### Usage

```
FDR(pvals, qllevel = 0.05)
```

### Arguments

<code>pvals</code>	a vector of pvals on which to conduct the multiple testing
<code>qllevel</code>	the proportion of false positives desired

**Value**

fdr.id            NULL if no significant tests, or the index of the maximal p-value satisfying the FDR condition.

**Note**

This is used primarily in Chapter 7.

**Source**

Built off of <https://www.stat.berkeley.edu/~paciorek/code/fdr/fdr.R>.

**References**

You can find demonstrations of astsa capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

---

ffbs

*Forward Filtering Backward Sampling*


---

**Description**

FFBS algorithm for state space models

**Usage**

```
ffbs(y, A, mu0, Sigma0, Phi, sQ, sR, Ups = NULL, Gam = NULL, input = NULL)
```

**Arguments**

y	Data matrix, vector or time series.
A	Observation matrix. Can be constant or an array with $\text{dim}=c(q,p,n)$ if time varying.
mu0	Initial state mean.
Sigma0	Initial state covariance matrix.
Phi	State transition matrix.
sQ	State error covariance matrix is $Q = \text{sQ} \% \% \text{t}(\text{sQ})$ – see details below. In the univariate case, it is the standard deviation.
sR	Observation error covariance matrix is $R = \text{sR} \% \% \text{t}(\text{sR})$ – see details below. In the univariate case, it is the standard deviation.
Ups	State input matrix.
Gam	Observation input matrix.
input	matrix or vector of inputs having the same row dimension as y.

## Details

For a linear state space model, the FFBS algorithm provides a way to sample a state sequence  $x_{0:n}$  from the posterior  $\pi(x_{0:n} \mid \Theta, y_{1:n})$  with parameters  $\Theta$  and data  $y_{1:n}$ .

The general model is

$$x_t = \Phi x_{t-1} + \Upsilon u_t + sQ w_t \quad w_t \sim iid N(0, I)$$

$$y_t = A_t x_{t-1} + \Gamma u_t + sR v_t \quad v_t \sim iid N(0, I)$$

where  $w_t \perp v_t$ . Consequently the state noise covariance matrix is  $Q = sQ sQ'$  and the observation noise covariance matrix is  $R = sR sR'$  and  $sQ, sR$  do not have to be square as long as everything is conformable.

$x_t$  is p-dimensional,  $y_t$  is q-dimensional, and  $u_t$  is r-dimensional. Note that  $sQ w_t$  has to be p-dimensional, but  $w_t$  does not, and  $sR v_t$  has to be q-dimensional, but  $v_t$  does not.

## Value

Xs	An array of sampled states
X0n	The sampled initial state (because R is 1-based)

## Note

The script uses `Kfilter`. If  $A_t$  is constant wrt time, it is not necessary to input an array; see the example. The example below is just one pass of the algorithm; see the example at [FUN WITH ASTSA](#) for the real fun.

## Author(s)

D.S. Stoffer

## Source

Chapter 6 of the Shumway and Stoffer Springer text.

## References

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

## Examples

```
## Not run:

## -- this is just one pass --##
# generate some data
set.seed(1)
sQ = 1; sR = 3; n = 100
mu0 = 0; Sigma0 = 10; x0 = rnorm(1,mu0,Sigma0)
w = rnorm(n); v = rnorm(n)
x = c(x0 + sQ*w[1]); y = c(x[1] + sR*v[1]) # initialize
for (t in 2:n){
  x[t] = x[t-1] + sQ*w[t]
  y[t] = x[t] + sR*v[t]
}

## run one pass of FFBS, plot data, states and sampled states
run = ffbs(y, A=1, mu0=0, Sigma0=10, Phi=1, sQ=1, sR=3)
tsplot(cbind(y,run$Xs), spaghetti=TRUE, type='o', col=c(8,4), pch=c(1,NA))
legend('topleft', legend=c("y(t)","xs(t)"), lty=1, col=c(8,4), bty="n", pch=c(1,NA))

## End(Not run)
```

---

flu

*Monthly pneumonia and influenza deaths in the U.S., 1968 to 1978.*

---

## Description

Monthly pneumonia and influenza deaths per 10,000 people in the United States for 11 years, 1968 to 1978.

## Usage

```
data(flu)
```

## Format

The format is: Time-Series [1:132] from 1968 to 1979: 0.811 0.446 0.342 0.277 0.248 ...

## References

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

---

`fmri`*fMRI - complete data set*

---

## Description

Data (as a vector list) from an fMRI experiment in pain, listed by location and stimulus. The data are BOLD signals when a stimulus was applied for 32 seconds and then stopped for 32 seconds. The signal period is 64 seconds and the sampling rate was one observation every 2 seconds for 256 seconds ( $n = 128$ ). The number of subjects under each condition varies.

## Details

The LOCATIONS of the brain where the signal was measured were [1] Cortex 1: Primary Somatosensory, Contralateral, [2] Cortex 2: Primary Somatosensory, Ipsilateral, [3] Cortex 3: Secondary Somatosensory, Contralateral, [4] Cortex 4: Secondary Somatosensory, Ipsilateral, [5] Caudate, [6] Thalamus 1: Contralateral, [7] Thalamus 2: Ipsilateral, [8] Cerebellum 1: Contralateral and [9] Cerebellum 2: Ipsilateral.

The TREATMENTS or stimuli (and number of subjects in each condition) are [1] Awake-Brush (5 subjects), [2] Awake-Heat (4 subjects), [3] Awake-Shock (5 subjects), [4] Low-Brush (3 subjects), [5] Low-Heat (5 subjects), and [6] Low-Shock (4 subjects). Issue the command `summary(fmri)` for further details. In particular, awake (Awake) or mildly anesthetized (Low) subjects were subjected levels of periodic brushing (Brush), application of heat (Heat), and mild shock (Shock) effects.

As an example, `fmri$LT6` (Location 1, Treatment 6) will show the data for the four subjects receiving the Low-Shock treatment at the Cortex 1 location; note that `fmri[[6]]` will display the same data.

## Source

Joseph F. Antognini, Michael H. Buonocore, Elizabeth A. Disbrow, Earl Carstens, Isoflurane anesthesia blunts cerebral responses to noxious and innocuous stimuli: a fMRI study, *Life Sciences*, Volume 61, Issue 24, 1997, Pages PL349-PL354, ISSN 0024-3205, [https://doi.org/10.1016/S0024-3205\(97\)00960-0](https://doi.org/10.1016/S0024-3205(97)00960-0).

## References

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

---

`fmri1`*fMRI Data Used in Chapter 1*

---

**Description**

A data frame that consists of average fMRI BOLD signals at eight locations.

**Usage**

```
data(fmri1)
```

**Format**

The format is: mts [1:128, 1:9]

**Details**

Multiple time series consisting of fMRI BOLD signals at eight locations (in columns 2-9, column 1 is time period), when a stimulus was applied for 32 seconds and then stopped for 32 seconds. The signal period is 64 seconds and the sampling rate was one observation every 2 seconds for 256 seconds ( $n = 128$ ). The columns are labeled: "time" "cort1" "cort2" "cort3" "cort4" "thal1" "thal2" "cere1" "cere2".

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**See Also**

[fmri](#)

**Examples**

```
tsplot(fmri1[,2:9], ncolm=4, gg=TRUE, col=c(rep(4,4),rep(3,2),rep(6,2)))
```

---

gas

*Gas Prices*

---

### Description

New York Harbor conventional regular gasoline weekly spot price FOB (in cents per gallon) from 2000 to mid-2010.

### Format

The format is: Time-Series [1:545] from 2000 to 2010: 70.6 71 68.5 65.1 67.9 ...

### Details

Pairs with series oil

### Source

Data were obtained from: [https://www.eia.gov/dnav/pet/pet\\_pri\\_spt\\_s1\\_w.htm](https://www.eia.gov/dnav/pet/pet_pri_spt_s1_w.htm)

### References

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

### See Also

[oil](#)

### Examples

```
tspplot(cbind(gas,oil), spag=TRUE, col=2*1:2, addLegend=TRUE)
```

gdp

*Quarterly U.S. GDP*

---

**Description**

Seasonally adjusted quarterly U.S. GDP from 1947(1) to 2018(3).

**Format**

The format is: Time-Series [1:287] from 1947 to 2018: 2033 2028 2023 2055 2086 ...

**Source**

<https://tradingeconomics.com/united-states/gdp>

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**See Also**

[GDP](#), [GNP](#), [gnp](#)

---

GDP23*Quarterly U.S. GDP - updated to 2023*

---

**Description**

Seasonally adjusted quarterly U.S. GDP from 1947(1) to 2023(1).

**Format**

The format is: Time-Series [1:305] from 1947 to 2023: 243.164 245.968 249.585 259.745 ...

**Source**

<https://fred.stlouisfed.org/series/GDP>

## References

You can find demonstrations of astsa capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

## See Also

[gdp](#), [GNP](#), [gnp](#)

---

gnp

*Quarterly U.S. GNP*

---

## Description

Seasonally adjusted quarterly U.S. GNP from 1947(1) to 2002(3).

## Format

The format is: Time-Series [1:223] from 1947 to 2002: 1489 1497 1500 1524 1547 ...

## Source

<https://fred.stlouisfed.org/series/GNP>

## References

You can find demonstrations of astsa capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

## See Also

[GNP](#), [GDP](#), [gdp](#)

---

GNP23

*Quarterly U.S. GNP - updated to 2023*

---

### Description

Seasonally adjusted quarterly U.S. GNP from 1947(1) to 2003(1).

### Format

The format is: Time-Series [1:305] from 1947 to 2023: 244.142 247.063 250.716 260.981 ...

### Source

<https://fred.stlouisfed.org/series/GNP>

### References

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

### See Also

[gnp](#), [GDP](#), [gdp](#)

---

Grid

*A Better Add Grid to a Plot*

---

### Description

Adds a grid to an existing plot with major and minor ticks. Works like R graphics `grid()` but the grid lines are solid and gray and minor ticks are produced by default.

### Usage

```
Grid(nx = NULL, ny = nx, col = gray(0.9), lty = 1, lwd = par("lwd"), equilogs = TRUE,
     minor = TRUE, nxm = 2, nym = 2, tick.ratio = 0.5, xm.grid = TRUE, ym.grid = TRUE, ...)
```

**Arguments**

<code>nx, ny</code>	number of cells of the grid in x and y direction. When NULL, as per default, the grid aligns with the tick marks on the corresponding default axis (i.e., tickmarks as computed by <code>axTicks</code> ). When NA, no grid lines are drawn in the corresponding direction.
<code>col</code>	color of the grid lines.
<code>lty</code>	line type of the grid lines.
<code>lwd</code>	line width of the grid lines.
<code>equilogs</code>	logical, only used when log coordinates and alignment with the axis tick marks are active. Setting <code>equilogs = FALSE</code> in that case gives non equidistant tick aligned grid lines.
<code>minor</code>	logical with TRUE (default) adding minor ticks.
<code>nxm, nym</code>	number of intervals in which to divide the area between major tick marks on the x-axis (y-axis). If <code>minor=TRUE</code> , should be > 1 or no minor ticks will be drawn.
<code>tick.ratio</code>	ratio of lengths of minor tick marks to major tick marks. The length of major tick marks is retrieved from <code>par("tck")</code> .
<code>xm.grid, ym.grid</code>	if TRUE (default), adds grid lines at minor x-axis, y-axis ticks.
<code>...</code>	other graphical parameters;

**Author(s)**

D.S. Stoffer

**Source**

The code for `grid()` in R 'graphics' and `minor.tick()` from the 'Hmisc' package were combined and then washed, polished, and coated with a subtle metallic finish. The grid now sparkles in the sunlight.

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**See Also**

[grid](#)

---

`gtemp.month`*Monthly global average surface temperatures by year*

---

### Description

Monthly global average surface temperatures by year. The temperature of the air measured 2 meters above the ground, encompassing land, sea, and in-land water surfaces.

### Format

A data frame with 12 monthly observations (as rows) for the years 1975-2023 (as columns in reverse order).

### Details

Temperature of air at 2m above the surface of land, sea or in-land waters. 2m temperature is calculated by interpolating between the lowest model level and the Earth's surface, taking account of the atmospheric conditions. Technical details at <https://cds.climate.copernicus.eu/datasets/reanalysis-era5-pressure>

### Source

<https://ourworldindata.org/grapher/monthly-average-surface-temperatures-by-year>

### References

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

### Examples

```
# functional data plot showing global warming
tsplot(gtemp.month, spaghetti=TRUE, col=rainbow(49, start=.2, v=.8, rev=TRUE), ylab='\u00b0C',
       xlab='Month', xaxt='n', main='Mean Monthly Global Temperature', lwd=c(3,rep(1,47),3))
axis(1, labels=Months, at=1:12)
text(10, 13, '1975')
text(10.3, 15.5, '2023')
```

---

gtemp\_both

*Global mean land and open ocean temperature deviations, 1850-2023*

---

### Description

Annual temperature anomalies (in degrees centigrade) averaged over the Earth's land and ocean area from 1850 to 2023. Anomalies are with respect to the 1991-2020 average.

### Format

The format is: Time-Series [1:174] from 1850 to 2023: -0.24 -0.25 -0.27 -0.15 -0.05 -0.16 -0.29 -0.32 -0.19 -0.04 ...

### Source

<https://www.ncei.noaa.gov/access/monitoring/climate-at-a-glance/global/time-series/>

### References

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

### See Also

[gtemp\\_land](#), [gtemp\\_ocean](#)

---

gtemp\_land

*Global mean land temperature deviations, 1850-2023*

---

### Description

Annual temperature anomalies (in degrees centigrade) averaged over the Earth's land area from 1850 to 2023. Anomalies are with respect to the 1991-2020 average.

### Format

The format is: Time-Series [1:174] from 1850 to 2023: -0.50 -0.60 -0.50 -0.50 -0.20 -0.50 -0.80 -0.40 -0.40 -0.10 ...

### Source

<https://www.ncei.noaa.gov/access/monitoring/climate-at-a-glance/global/time-series/>

## References

You can find demonstrations of astsa capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

## See Also

[gtemp\\_ocean](#), [gtemp\\_both](#)

---

gtemp\_ocean

*Global mean ocean temperature deviations, 1850-2023*

---

## Description

Annual sea surface temperature anomalies averaged over the part of the ocean that is free of ice at all times (open ocean) from 1850 to 2023. Anomalies are with respect to the 1991-2020 average.

## Format

The format is: Time-Series [1:174] from 1850 to 2023: -0.12 -0.08 -0.14 0.04 0.04 0.00 -0.05 -0.27 -0.09 0.01 ...

## Source

<https://www.ncei.noaa.gov/access/monitoring/climate-at-a-glance/global/time-series/>

## References

You can find demonstrations of astsa capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

## See Also

[gtemp\\_land](#), [gtemp\\_both](#)

---

Hare

*Snowshoe Hare*

---

### Description

This is one of the classic studies of predator-prey interactions, the 90-year data set is the number, in thousands, of snowshoe hare pelts purchased by the Hudson's Bay Company (HBC) of Canada. While this is an indirect measure of predation, the assumption is that there is a direct relationship between the number of pelts collected and the number of hare and lynx in the wild.

If you are interested in the brutal story of HBC, see the video [https://www.nfb.ca/film/other\\_side\\_of\\_the\\_ledger](https://www.nfb.ca/film/other_side_of_the_ledger) or read the article <https://canadiangeographic.ca/articles/the-untold-story-of-the-hudsons-bay-company>.

### Format

The format is: Time-Series [1:91] from 1845 to 1935: 19.6 19.6 19.6 12 28 ...

### Note

This data set pairs with [Lynx](#). The data are in units of one thousand.

### Source

From Odum's "Fundamentals of Ecology", p. 191. Data listed at:  
<http://people.whitman.edu/~hundredr/courses/M250F03/M250.html>  
scroll down to: Chapter 6, Difference Equations.

NB: For some reason, there is not a secure and encrypted version of the site above.

### References

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

### See Also

[Lynx](#)

---

HCT *Hematocrit Levels*

---

**Description**

HCT: Measurements made for 91 days on the three variables, log(white blood count) [WBC], log(platelet) [PLT] and hematocrit [HCT]. Missing data code is 0 (zero).

**Format**

The format is: Time-Series [1:91] from 1 to 91: 30 30 28.5 34.5 34 32 30.5 31 33 34 ...

**Details**

See Examples 6.1 and 6.9 for more details.

**Source**

Jones, R.H. (1984). Fitting multivariate models to unequally spaced data. In *Time Series Analysis of Irregularly Observed Data*, pp. 158-188. E. Parzen, ed. Lecture Notes in Statistics, 25, New York: Springer-Verlag.

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**See Also**

[blood](#), [PLT](#), [WBC](#)

---

hor *Hawaiian occupancy rates*

---

**Description**

Quarterly Hawaiian hotel occupancy rate (percent of rooms occupied) from 1982-I to 2015-IV

**Format**

The format is: Time-Series [1:136] from 1982 to 2015: 79 65.9 70.9 66.7 ...

**Source**

<https://dbedt.hawaii.gov/economic/qser/tourism/>

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**Examples**

```
tsplot(hor, type='c') # plot data and
text(hor, labels=1:4, col=c(1,4,2,6), cex=.9) # add quarter labels
```

---

 jj

---

*Johnson and Johnson Quarterly Earnings Per Share*


---

**Description**

Johnson and Johnson quarterly earnings per share, 84 quarters (21 years) measured from the first quarter of 1960 to the last quarter of 1980.

**Format**

The format is: Time-Series [1:84] from 1960 to 1981: 0.71 0.63 0.85 0.44 0.61 0.69 0.92 0.55 0.72 0.77 ...

**Details**

The data were provided (personal communication) by Professor Paul Griffin, <https://gsm.ucdavis.edu/profile/paul-g> of the Graduate School of Management, University of California, Davis. This data set is also included with the R distribution as `JohnsonJohnson`.

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

Kfilter

*Quick Kalman Filter***Description**

Returns both the predicted and filtered values for various linear state space models; it also evaluates the likelihood at the given parameter values.

**Usage**

```
Kfilter(y, A, mu0, Sigma0, Phi, sQ, sR, Ups = NULL, Gam = NULL,
        input = NULL, S = NULL, version = 1)
```

**Arguments**

y	data matrix (n × q), vector or time series, n = number of observations. Use NA or zero (0) for missing data.
A	can be constant or an array with dimension dim=c(q, p, n) if time varying (see details). Use NA or zero (0) for missing data.
mu0	initial state mean vector (p × 1)
Sigma0	initial state covariance matrix (p × p)
Phi	state transition matrix (p × p)
sQ	state error pre-matrix (see details)
sR	observation error pre-matrix (see details)
Ups	state input matrix (p × r); leave as NULL (default) if not needed
Gam	observation input matrix (q × r); leave as NULL (default) if not needed
input	NULL (default) if not needed or a matrix (n × r) of inputs having the same row dimension (n) as y
S	covariance matrix between the (not premultiplied) state and observation errors; not necessary to specify if not needed and only used if version=2. See details for more information.
version	either 1 (default) or 2; version 2 allows for correlated errors

**Details**

This script replaces `Kfilter0`, `Kfilter1`, and `Kfilter2` by combining all cases. The major difference is how to specify the covariance matrices; in particular,  $sQ = t(cQ)$  and  $sR = t(cR)$  where  $cQ$  and  $cR$  were used in `Kfilter0-1-2` scripts.

The states  $x_t$  are  $p$ -dimensional, the data  $y_t$  are  $q$ -dimensional, and the inputs  $u_t$  are  $r$ -dimensional for  $t = 1, \dots, n$ . The initial state is  $x_0 \sim N(\mu_0, \Sigma_0)$ .

The measurement matrices  $A_t$  can be constant or time varying. If time varying, they should be entered as an array of dimension  $\text{dim} = c(q, p, n)$ . Otherwise, just enter the constant value making sure it has the appropriate  $q \times p$  dimension.

Version 1 (default): The general model is

$$x_t = \Phi x_{t-1} + \Upsilon u_t + sQ w_t \quad w_t \sim iid N(0, I)$$

$$y_t = A_t x_{t-1} + \Gamma u_t + sR v_t \quad v_t \sim iid N(0, I)$$

where  $w_t \perp v_t$ . Consequently the state noise covariance matrix is  $Q = sQ sQ'$  and the observation noise covariance matrix is  $R = sR sR'$  and  $sQ, sR$  do not have to be square as long as everything is conformable. Notice the specification of the state and observation covariances has changed from the original scripts.

*NOTE:* If it is easier to model in terms of  $Q$  and  $R$ , simply input the square root matrices  $sQ = Q^{.5}$  and  $sR = R^{.5}$ .

Version 2 (correlated errors): The general model is

$$x_{t+1} = \Phi x_t + \Upsilon u_{t+1} + sQ w_t \quad w_t \sim iid N(0, I)$$

$$y_t = A_t x_{t-1} + \Gamma u_t + sR v_t \quad v_t \sim iid N(0, I)$$

where  $S = \text{Cov}(w_t, v_t)$ , and NOT  $\text{Cov}(sQ w_t, sR v_t)$ .

*NOTE:* If it is easier to model in terms of  $Q$  and  $R$ , simply input the square root matrices  $sQ = Q^{.5}$  and  $sR = R^{.5}$ .

Note that in either version,  $sQ w_t$  has to be p-dimensional, but  $w_t$  does not, and  $sR v_t$  has to be q-dimensional, but  $v_t$  does not.

## Value

Time varying values are returned as arrays.

Xp	one-step-ahead prediction of the state
Pp	mean square prediction error
Xf	filter value of the state
Pf	mean square filter error
like	the negative of the log likelihood
innov	innovation series
sig	innovation covariances
Kn	last value of the gain, needed for smoothing

## Note

If it is easier to model in terms of  $Q$  and  $R$ , simply input the square root matrices  $sQ = Q^{.5}$  and  $sR = R^{.5}$ .

## Author(s)

D.S. Stoffer

## References

You can find demonstrations of astsa capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

## See Also

[Ksmooth](#)

## Examples

```
# generate some data
set.seed(1)
sQ = 1; sR = 3; n = 100
mu0 = 0; Sigma0 = 10; x0 = rnorm(1,mu0,Sigma0)
w = rnorm(n); v = rnorm(n)
x = c(x0 + sQ*w[1]); y = c(x[1] + sR*v[1]) # initialize
for (t in 2:n){
  x[t] = x[t-1] + sQ*w[t]
  y[t] = x[t] + sR*v[t]
}
# run and plot the filter
run = Kfilter(y, A=1, mu0, Sigma0, Phi=1, sQ, sR)

tspplot(cbind(x, y, Xf=run$Xf), spaghetti=TRUE, type='o', col=c(3,4,6), pch=c(NA,1,NA),
        addLegend=TRUE, location='topleft', lwd=c(2,1,2))
```

---

Ksmooth

*Quick Kalman Smoother*

---

## Description

Returns the smoother values for various linear state space models. The predicted and filtered values and the likelihood at the given parameter values are also returned (via `Kfilter`).

## Usage

```
Ksmooth(y, A, mu0, Sigma0, Phi, sQ, sR, Ups = NULL, Gam = NULL,
        input = NULL, S = NULL, version = 1)
```

**Arguments**

y	data matrix (n x q), vector or time series, n = number of observations. Use NA or zero (0) for missing data.
A	can be constant or an array with dimension dim=c(q,p,n) if time varying (see details). Use NA or zero (0) for missing data.
mu0	initial state mean vector (p x 1)
Sigma0	initial state covariance matrix (p x p)
Phi	state transition matrix (p x p)
sQ	state error pre-matrix (see details)
sR	observation error pre-matrix (see details)
Ups	state input matrix (p x r); leave as NULL (default) if not needed
Gam	observation input matrix (q x r); leave as NULL (default) if not needed
input	NULL (default) if not needed or a matrix (n x r) of inputs having the same row dimension (n) as y
S	covariance matrix between state and observation errors; not necessary to specify if not needed and only used if version=2; see details
version	either 1 (default) or 2; version 2 allows for correlated errors

**Details**

The states  $x_t$  are p-dimensional, the data  $y_t$  are q-dimensional, and the inputs  $u_t$  are r-dimensional for  $t = 1, \dots, n$ . The initial state is  $x_0 \sim N(\mu_0, \Sigma_0)$ .

The measurement matrices  $A_t$  can be constant or time varying. If time varying, they should be entered as an array of dimension dim = c(q,p,n). Otherwise, just enter the constant value making sure it has the appropriate  $q \times p$  dimension.

Version 1 (default): The general model is

$$\begin{aligned}x_t &= \Phi x_{t-1} + \Upsilon u_t + sQ w_t \quad w_t \sim iid N(0, I) \\y_t &= A_t x_{t-1} + \Gamma u_t + sR v_t \quad v_t \sim iid N(0, I)\end{aligned}$$

where  $w_t \perp v_t$ . Consequently the state noise covariance matrix is  $Q = sQ sQ'$  and the observation noise covariance matrix is  $R = sR sR'$  and  $sQ, sR$  do not have to be square as long as everything is conformable. Notice the specification of the state and observation covariances has changed from the original scripts.

**NOTE:** If it is easier to model in terms of  $Q$  and  $R$ , simply input the square root matrices  $sQ = Q^{.5}$  and  $sR = R^{.5}$ .

Version 2 (correlated errors): The general model is

$$\begin{aligned}x_{t+1} &= \Phi x_t + \Upsilon u_{t+1} + sQ w_t \quad w_t \sim iid N(0, I) \\y_t &= A_t x_{t-1} + \Gamma u_t + sR v_t \quad v_t \sim iid N(0, I)\end{aligned}$$

where  $S = Cov(w_t, v_t)$ , and NOT  $Cov(sQ w_t, sR v_t)$ .

**NOTE:** If it is easier to model in terms of  $Q$  and  $R$ , simply input the square root matrices  $sQ = Q^{.5}$  and  $sR = R^{.5}$ .

Note that in either version,  $sQ w_t$  has to be p-dimensional, but  $w_t$  does not, and  $sR v_t$  has to be q-dimensional, but  $v_t$  does not.

**Value**

Time varying values are returned as arrays.

Xs	state smoothers
Ps	smoother mean square error
X0n	initial mean smoother
P0n	initial smoother covariance
J0	initial value of the J matrix
J	the J matrices
Xp	state predictors
Pp	mean square prediction error
Xf	state filters
Pf	mean square filter error
like	negative of the log likelihood
innov	innovation series
sig	innovation covariances
Kn	the value of the last Gain

**Author(s)**

D.S. Stoffer

**References**

You can find demonstrations of astsa capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**See Also**

[Kfilter](#)

**Examples**

```
# generate some data
set.seed(1)
sQ = 1; sR = 3; n = 100
mu0 = 0; Sigma0 = 10; x0 = rnorm(1,mu0,Sigma0)
w = rnorm(n); v = rnorm(n)
x = c(x0 + sQ*w[1]); y = c(x[1] + sR*v[1]) # initialize
for (t in 2:n){
  x[t] = x[t-1] + sQ*w[t]
```

```

    y[t] = x[t] + sR*v[t]
  }

# run and plot the smoother
run = Ksmooth(y, A=1, mu0, Sigma0, Phi=1, sQ, sR)

tsplot(cbind(x, y, Xs=run$Xs), spaghetti=TRUE, type='o', col=c(3,4,6), pch=c(NA,1,NA),
       addLegend=TRUE, location='topleft', lwd=c(2,1,2), gg=TRUE)

```

lag1.plot

*Lag Plot - one time series***Description**

Produces a grid of scatterplots of a series versus lagged values of the series.

**Usage**

```
lag1.plot(series, max.lag = 1, corr = TRUE, smooth = TRUE, col = gray(.1), bg = NA,
         lw1 = 1, lwc = 2, bg1 = NULL, ltcol = 1, box.col = NULL, cex = .9,
         gg = FALSE, location="topright", xname = NULL, main = NULL, ...)
```

**Arguments**

series	the data
max.lag	maximum lag
corr	if TRUE, shows the autocorrelation value in a legend
smooth	if TRUE, adds a lowess fit to each scatterplot
col	color of points; default is gray(.1)
bg	background color for filled plot characters
lw1	width of lowess line; default is 1
lwc	color of lowess line; default is 2 (red)
bg1	background of the ACF legend; default is semitransparent
ltcol	legend text color; default is black
box.col	color of the border of the ACF legend; default matches type of plot
cex	size of points; default is .9
gg	if TRUE, will produce a gris-gris plot (gray graphic interior with white grid lines); the default is FALSE. The grammar of astsa is voodoo
location	the location of the ACF legend with options "bottomright", "bottom", "bottomleft", "left", "topleft", "top", "topright" (the default), "right" and "center".
xname	a string; name of the series to be used for axis labels. If NULL, the name of the series as input is used.
main	a string for the title if desired, otherwise there is no title.
...	additional graphical arguments

**Author(s)**

D.S. Stoffer

**References**

You can find demonstrations of astsa capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**See Also**

[lag2.plot](#), [tspairs](#)

**Examples**

```
lag1.plot(log(varve), max.lag=9, lwl=3, lwc=6, col=5, location='topleft')
```

```
lag1.plot(soi, 12, pch=19, col=astsa.col(4, .3), gg=TRUE)
```

```
lag1.plot(diff(log(varve)), 4, xname='V', col=6, cex=1.1, lwc=5, pch=5,
          main=bquote(V(t)==nabla*log(varve[t])))
```

---

lag2.plot

*Lag Plot - two time series*

---

**Description**

Produces a grid of scatterplots of one series versus another lagged. The first named series is the one that gets lagged.

**Usage**

```
lag2.plot(series1, series2, max.lag = 0, corr = TRUE, smooth = TRUE, col = gray(.1),
          bg = NA, lwl = 1, lwc = 2, bgl = NULL, ltcoll = 1, box.col = NULL, cex = .9,
          gg = FALSE, location="topright", xname=NULL, yname=NULL, main=NULL, ...)
```

**Arguments**

series1	first series (the one that gets lagged)
series2	second series
max.lag	maximum number of lags
corr	if TRUE, shows the cross-correlation value in a legend
smooth	if TRUE, adds a lowess fit to each scatterplot

col	color of points; default is gray(.1)
bg	background color for filled plot characters
lwl	width of lowess line; default is 1
lwc	color of lowess line; default is 2 (red)
bg1	background of the ACF legend; default is semitransparent
ltcol	legend text color; default is black
box.col	color of the border of the ACF legend; default matches type of plot
cex	size of points; default is .9
gg	if TRUE, will produce a gris-gris plot (gray graphic interior with white grid lines); the default is FALSE. The grammar of astsa is voodoo
location	the location of the CCF legend with options "bottomright", "bottom", "bottomleft", "left", "topleft", "top", "topright" (the default), "right" and "center".
xname, yname	strings for the names of the series to be used for axis labels. If NULL, the name of the series as input is used.
main	a string for the title if desired, otherwise there is no title.
...	additional graphical parameters

**Author(s)**

D.S. Stoffer

**References**

You can find demonstrations of astsa capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**See Also**

[lag1.plot](#), [tspairs](#)

**Examples**

```
lag2.plot(Hare, Lynx, max.lag=5, lwl=2, lwc=3, cex=1.5, pch=24, bg='orange')
```

```
lag2.plot(soi, rec, 8, cex=1.1, pch=19, col=5, lwl=2, location="bottomleft")
```

```
lag2.plot(diff(lead), diff(sales), 5, smooth=FALSE, gg=TRUE, pch=8, col=6,
          xname='dL', yname='dS', main='BJ Sales')
```

LagReg

*Lagged Regression***Description**

Performs lagged regression as discussed in Chapter 4.

**Usage**

```
LagReg(input, output, L = c(3, 3), M = 40, threshold = 0,
       inverse = FALSE)
```

**Arguments**

input	input series
output	output series
L	degree of smoothing; see spans in the help file for <code>spec.pgram</code> .
M	must be even; number of terms used in the lagged regression
threshold	the cut-off used to set small (in absolute value) regression coefficients equal to zero
inverse	if TRUE, will fit a forward-lagged regression

**Details**

For a bivariate series, `input` is the input series and `output` is the output series. The degree of smoothing for the spectral estimate is given by `L`; see spans in the help file for `spec.pgram`. The number of terms used in the lagged regression approximation is given by `M`, which must be even. The threshold value is the cut-off used to set small (in absolute value) regression coefficients equal to zero (it is easiest to run `LagReg` twice, once with the default threshold of zero, and then again after inspecting the resulting coefficients and the corresponding values of the CCF). Setting `inverse=TRUE` will fit a forward-lagged regression; the default is to run a backward-lagged regression. The script is based on code that was contributed by Professor Doug Wiens, Department of Mathematical and Statistical Sciences, University of Alberta.

**Value**

Graphs of the estimated impulse response function, the CCF, and the output with the predicted values superimposed.

beta	Estimated coefficients
fit	The output series, the fitted values, and the residuals

**Note**

See Chapter 4 of the text for an example.

**Author(s)**

D.S. Stoffer

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

---

 lap

---

*LA Pollution-Mortality Study*


---

**Description**

LA Pollution-Mortality Study (1970-1979), weekly data.

**Format**

The format is: `mts [1:508, 1:11]`

**Details**

columns are time series	with names
(1) Total Mortality	<code>tmort</code>
(2) Respiratory Mortality	<code>rmort</code>
(3) Cardiovascular Mortality	<code>cmort</code>
(4) Temperature	<code>tempr</code>
(5) Relative Humidity	<code>rh</code>
(6) Carbon Monoxide	<code>co</code>
(7) Sulfur Dioxide	<code>so2</code>
(8) Nitrogen Dioxide	<code>no2</code>
(9) Hydrocarbons	<code>hycarb</code>
(10) Ozone	<code>o3</code>
(11) Particulates	<code>part</code>

**Note**

Details may be found in <http://www.sungpark.net/ShumwayAzariPawitan88.pdf>

## References

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

## See Also

[lap.xts](#)

---

lap.xts

*LA Pollution-Mortality Study: Sampled Daily*

---

## Description

Original data from a study of the effects of pollution and weather on mortality, LA, 1970-1979. These are 3652 daily observations for the 10 year period. The data set is an 'xts' object indexed by Date.

## Format

The format is: An xts object on 1970-01-01 / 1979-12-31 containing:

Data: double [3652, 11]

Columns: Tmort, Rmort, Cmort, Temp, Rhumid, CO, SO2, NO2, HC, Ozone, Part

Index: Date [3652] (TZ: "UTC")

## Details

columns are time series	with names
(1) Total Mortality	Tmort
(2) Respiratory Mortality	Rmort
(3) Cardiovascular Mortality	Cmort
(4) Temperature	Temp
(5) Relative Humidity	Rhumid
(6) Carbon Monoxide	CO
(7) Sulfur Dioxide	SO2
(8) Nitrogen Dioxide	NO2
(9) Hydrocarbons	HC
(10) Ozone	Ozone
(11) Particulates	Part

**Note**

These are the original data from <https://github.com/DSStoffer/dsstoffer.github.io/blob/main/files/LAP.pdf>.

The weekly data in `lap` were taken from this data set last century. The details, however, were never entirely made clear and it's too late to get them now. It is easy to pull out the weekly averages from this data set, and how to do so is given in the Examples section below; the resulting data set will be slightly different than `lap`. The names for this data set are different from `lap`, the main difference is these names have capitals.

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**See Also**

[lap](#)

**Examples**

```
# if 'xts' is not installed you can still do this
tsplot(timex(lap.xts), lap.xts, ncolm=3, col=astsa.col(4,wheel=TRUE,num=11), scale=.9)

# differencing (on its own) loses an obs
dCmort = diff(log(lap.xts[, 'Cmort']))
tsplot(timex(lap.xts)[-1], dCmort, col=4, gg=TRUE)

# classic IMA(1,1)
acf2(dCmort, col=2:7, lwd=4)

## Not run:

library(xts) # assumes package has been installed

plot(lap.xts$Cmort, col=4)

lapw = apply.weekly(lap.xts, FUN=colMeans) # get weekly averages

plot(lapw[,c('Cmort', 'Temp', 'Part')], col=astsa.col(2:4, .7), main=NA)
addLegend(col=2:4, lty=1, lwd=2, ncol=3, bty="white")

sarima(lapw$Cmort, 0,1,1, no.constant=TRUE) # fit ARIMA(0,1,1) to weekly Cmort

## End(Not run)
```

---

lead	<i>Leading Indicator</i>
------	--------------------------

---

**Description**

Leading indicator, 150 months; taken from Box and Jenkins (1970).

**Usage**

```
data(lead)
```

**Format**

The format is: Time-Series [1:150] from 1 to 150: 10.01 10.07 10.32 9.75 10.33 ...

**Details**

This is also the R time series `BJsales.lead`: The sales time series `BJsales` and leading indicator `BJsales.lead` each contain 150 observations. The objects are of class "ts".

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**See Also**

[sales](#)

---

Lynx	<i>Canadian Lynx</i>
------	----------------------

---

**Description**

This is one of the classic studies of predator-prey interactions, the 90-year data set is the number, in thousands, of lynx pelts purchased by the Hudson's Bay Company (HBC) of Canada. While this is an indirect measure of predation, the assumption is that there is a direct relationship between the number of pelts collected and the number of hare and lynx in the wild.

If you are interested in the brutal story of HBC, see the video [https://www.nfb.ca/film/other\\_side\\_of\\_the\\_ledger](https://www.nfb.ca/film/other_side_of_the_ledger) or read the article <https://canadiangeographic.ca/articles/the-untold-story-of-the-hudsons-bay-company>.

**Format**

The format is: Time-Series [1:91] from 1845 to 1935: 30.1 45.1 49.1 39.5 21.2 ...

**Note**

The data are in units of one thousand. This data set pairs with [Hare](#) and is NOT the same as [lynx](#).

**Source**

From Odum's "Fundamentals of Ecology", p. 191. Additional information at <http://people.whitman.edu/~hundlejr/courses/M250F03/M250.html>  
scroll down to: Chapter 6, Difference Equations

NB: For some reason, there is not a secure and encrypted version of the site above.

**References**

You can find demonstrations of astsa capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**See Also**

[Hare](#)

---

matrixpwr

*Powers of a Square Matrix*

---

**Description**

matrixpwr computes powers of a square matrix including negative powers for nonsingular matrices.  
%% is a more intuitive interface as an operator.

**Usage**

```
matrixpwr(A, power)
```

```
A %% power
```

**Arguments**

A	a square matrix
power	single numeric

**Details**

Raises matrix to the specified power. The matrix must be square and if power < 0, the matrix must be nonsingular.

Note that `%%` is defined as `"%%" <- function(A, power) matrixpwr(A, power)`

If power = 0, the identity matrix is returned.

**Value**

Returns matrix raised to the given power.

**Author(s)**

D.S. Stoffer

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**Examples**

```
# 2-state Markov transition matrix
( P = matrix(c(.7,.4,.3,.6), 2) )
# moving to steady state
P %% 5
P %% 10

# surround with parentheses if used in an expression
c(.2, .8) %% (P %% 50)

# Inverse square root
var(econ5) %% -.5
```

**Description**

Bimonthly MEI values, starting with Dec1949/Jan1950 through Oct/Nov2019. All values are normalized for each bimonthly season so that the 44 values from 1950 to 1993 have an average of zero and a standard deviation of 1. Larger values correspond to warmer temperatures (unlike `soi` and `ENSO`).

**Format**

The format is: Time-Series [1:827] from 1950 to 2019: -1.03 -1.13 -1.28 -1.07 -1.43 ...

**Details**

For full details, see <https://psl.noaa.gov/enso/mei.old/mei.html>. Multivariate ENSO Index (MEI) is a combined score on the six main observed variables over the tropical Pacific. These six variables are: sea-level pressure (P), zonal (U) and meridional (V) components of the surface wind, sea surface temperature (S), surface air temperature (A), and total cloudiness fraction of the sky (C). These observations have been collected and published in ICOADS for many years. The MEI is computed separately for each of twelve sliding bi-monthly seasons (Dec/Jan, Jan/Feb, ..., Nov/Dec). After spatially filtering the individual fields into clusters, the MEI is calculated as the first unrotated Principal Component (PC) of all six observed fields combined. This is accomplished by normalizing the total variance of each field first, and then performing the extraction of the first PC on the covariance matrix of the combined fields. In order to keep the MEI comparable, all seasonal values are standardized with respect to each season and to the 1950-93 reference period.

Weak El Nino: MEI is between +0.5 and +1.0.

Moderate El Nino: MEI is between +1.0 and +1.5.

Strong El Nino: MEI is between +1.5 and +2.0.

Very Strong El Nino: MEI is at or above +2.0.

Values below the negative of these indicate La Nina conditions.

**Source**

<https://psl.noaa.gov/enso/mei.old/table.html>

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**See Also**

[soi](#), [ENSO](#), [MEI2](#)

**Examples**

```
tspplot(cbind(MEI, MEI2), spag=TRUE, col=2*2:1, addLegend=TRUE, nym=2, gg=TRUE)
```

```
tspairs(ts.intersect(MEI, MEI2), location='top')
```

---

MEI2

*Multivariate El Nino/Southern Oscillation Index (version 2)*

---

### Description

Bimonthly MEI values (version 2), starting with Dec1978/Jan1979 through Apr/May2025. These data are similar to MEI and larger values correspond to warmer temperatures (unlike soi and ENSO).

### Format

The format is: Time-Series [1:557] from 1979 to 2025: 0.47 0.29 -0.05 0.21 0.27 -0.11 -0.11 0.47 0.38 0.23 ...

### Details

For details, see <https://www.psl.noaa.gov/enso/mei> and [MEI](#), which is version 1.

The key differences between MEI version 2 and version 1 are the input variables used, the source of the data, and the dates of the historical record. MEI version 2 relies on reanalysis data using modern numerical weather models to process and combine historical weather observations into a comprehensive, globally complete, and consistent dataset of past weather and climate and satellite data. This eliminates inconsistencies caused by relying on potentially less reliable ship observations, especially in earlier decades. The switch from cloud cover fraction to Outgoing Longwave Radiation (OLR) provides a more direct and accurate measurement of atmospheric convection, a critical component of ENSO. Despite the differences, MEI version 2 and the original MEI version 1 are very highly correlated for the overlapping period.

Weak El Nino: MEI is between +0.5 and +1.0.

Moderate El Nino: MEI is between +1.0 and +1.5.

Strong El Nino: MEI is between +1.5 and +2.0.

Very Strong El Nino: MEI is at or above +2.0.

Values below the negative of these indicate La Nina conditions.

### Source

<https://www.psl.noaa.gov/enso/mei>

### References

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

### See Also

[soi](#), [ENSO](#), [MEI](#)

**Examples**

```
tsplot(cbind(MEI, MEI2), spag=TRUE, col=2*2:1, addLegend=TRUE, nym=2, gg=TRUE)

tspairs(ts.intersect(MEI, MEI2), location='top')
```

---

Months

*Month Labels*

---

**Description**

Provides labels for the (English) months of the year to be used in plotting monthly time series.

**Format**

The format is: chr [1:12] "J" "F" "M" "A" "M" "J" "J" "A" "S" "O" "N" "D"

**Note**

Hi Kids. The months of the year in English are:

January, February, March, April, May, June, July, August, September, October, November, December.

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**Examples**

```
sAR = sarima.sim(sar=.9, S=12, n=36)
tsplot(sAR, type='c')
points(sAR, pch=Months, cex=1.1, font=4, col=1:4)
```

mvspec

*Univariate and Multivariate Spectral Estimation***Description**

This is `spec.pgram` with a number changes and written so you can easily extract the estimate of the multivariate spectral matrix as `fxx`. The bandwidth calculation has been changed to the more practical definition given in the text and this can be used to replace `spec.pgram`.

**Usage**

```
mvspec(x, spans = NULL, kernel = NULL, taper = 0, pad = 0, fast = TRUE,
       demean = FALSE, detrend = TRUE, lowess = FALSE, log = 'n', plot = TRUE,
       gg = FALSE, type = NULL, na.action = na.fail, nxm = 2, nym = 1, main = NULL,
       xlab = NULL, cex.main = NULL, ci = .95, ci.col = 4, plot.type, report=TRUE, ...)
```

**Arguments**

<code>x</code>	univariate or multivariate time series (i.e., the <code>p</code> columns of <code>x</code> are time series)
<code>spans</code>	vector of odd integers giving the widths of modified Daniell smoothers to be used to smooth the periodogram
<code>kernel</code>	alternatively, a kernel smoother of class <code>tskernel</code>
<code>taper</code>	specifies the proportion of data to taper using a split cosine bell taper (.5 specifies a full taper)
<code>pad</code>	proportion of data to pad (zeros are added to the end of the series to increase its length by the proportion <code>pad</code> )
<code>fast</code>	logical; if <code>TRUE</code> , pad the series to a highly composite length
<code>demean</code>	if <code>TRUE</code> , series is demeaned first
<code>detrend</code>	if <code>TRUE</code> , series is detrended first (unless <code>demean</code> is <code>TRUE</code> )
<code>lowess</code>	if <code>TRUE</code> and <code>detrend</code> <code>TRUE</code> (and <code>demean</code> <code>FALSE</code> ), series is detrended using <code>lowess</code> first
<code>log</code>	if <code>log='y'</code> , spectra plotted on a log scale; otherwise a log scale is not used
<code>plot</code>	plot the estimated spectra
<code>gg</code>	if <code>TRUE</code> , will produce a gris-gris plot (gray graphic interior with white grid lines); the default is <code>FALSE</code> . The grammar of <code>astsa</code> is <code>voodoo</code>
<code>type</code>	type of plot to be drawn, defaults to lines (see <code>par</code> )
<code>na.action</code>	how to handle missing values
<code>nxm, nym</code>	the number of minor tick mark divisions on x-axis, y-axis; the default is one minor tick on the x-axis and none on the y-axis
<code>main</code>	title of the graphics; if <code>NULL</code> (default), a totally awesome title is generated dude, but if <code>NA</code> there will be no gnarly title and the top margin will be used for the plot

xlab	label for frequency axis; if NULL (default), a totally awesome label is generated for your viewing pleasure
cex.main	magnification for main title; default is 1.
ci	confidence level if one is drawn.
ci.col	color of the confidence interval if one is drawn.
plot.type	plot type for multivariate time series; leave blank or specify 'marginal' if a marginal plot of the spectra is desired, or set plot.type='coh' for coherency plot(s), or plot.type='phase' for phase plot(s).
report	if TRUE (default), prints bandwidth, degrees of freedom, and amount of tapering to screen
...	additional graphical arguments.

### Details

This is built off of `spec.pgram` from the `stats` package with a few changes in the defaults and written so you can easily extract the estimate of the multivariate spectral matrix as `fxx`.

The default for the plot is NOT to plot on a log scale and the graphic will have a grid. Overall, the graphics have been improved.

The bandwidth calculation has been changed to the more practical definition given in the text,  $(L_h/n.used) * frequency(x)$ . Also, the bandwidth is not displayed in the graphic, but is returned.

Although initially meant to be used to easily obtain multivariate (mv) spectral (spec) estimates, this script can be used for univariate time series as a replacement for `spec.pgram`.

Note that the script does not taper by default (`taper=0`); this forces the user to do "conscious tapering".

In the multivariate case (more than 2 series), if "coherency" or "phase" plots are desired, the result is a grid of plots. There is a "scale" factor that can be set to prevent the labels from getting too small if there are many series. The default is `scale=1`, and to expand the labels by 10% for example, set `scale=1.1`. Also, if there are many series, having minor tick marks can make the graphic look crowded; in this case, set `minor=FALSE`. Finally, the plots include a type of legend that shows the axes with their corresponding labels (by default). To turn the legend off, include `addLegend=FALSE` in the call. Again, these considerations only work in the multivariate case (more than 2 series).

### Value

All results are returned invisibly.

If `plot` is TRUE and smoothing is used, the bandwidth, degrees of freedom, and taper amount are printed.

An object of class "spec", which is a list containing at least the following components:

<code>fxx</code>	spectral matrix estimates; an array of dimensions <code>dim = c(p,p,nfreq)</code> .
<code>freq</code>	vector of frequencies at which the spectral density is estimated.
<code>spec</code>	vector (for univariate series) or matrix (for multivariate series) of estimates of the spectral density at frequencies corresponding to <code>freq</code> .
<code>details</code>	matrix with columns: frequency, period, spectral ordinate(s)

coh	NULL for univariate series. For multivariate time series, a matrix containing the squared coherency between different series. Column $i + (j - 1) * (j - 2)/2$ of coh contains the squared coherency between columns $i$ and $j$ of $x$ , where $i < j$ .
phase	NULL for univariate series. For multivariate time series a matrix containing the cross-spectrum phase between different series. The format is the same as coh.
Lh	Number of frequencies (approximate) used in the band.
n.used	Sample length used for the FFT
df	Degrees of freedom (may be approximate) associated with the spectral estimate.
bandwidth	Bandwidth (may be approximate) associated with the spectral estimate.
method	The method used to calculate the spectrum.

## References

You can find demonstrations of astsa capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

## Examples

```
# real raw periodogram
mvspec(soi)
mvspec(soi, log='y') # on a log scale

# smooth and some details printed
mvspec(soi, spans=c(7,7), taper=.1)$details[1:45,]

# bivariate example
deth = cbind(mdeaths, fdeaths) # two R data sets, male/female monthly deaths ...
tsplot(deth, type='b', col=c(4,6), spaghetti=TRUE, pch=c('M','F'), addLegend=TRUE)
dog = mvspec(deth, spans=c(3,3), taper=.1)
dog$fx[,1:5] # look at a few spectral matrix estimates
dog$bandwidth # bandwidth with time unit = year
dog$df # degrees of freedom
mvspec(deth, spans=c(3,3), taper=.1, plot.type='coh') # coherence

# multivariate example
mvspec(diff(log(econ5)), spans=c(5,5), col=5, lwd=2, ci=NA, gg=TRUE, minor=FALSE,
        plot.type='coh')

mvspec(diff(log(econ5)), spans=c(5,5), col=5, lwd=2, ci=NA, gg=TRUE, minor=FALSE,
        plot.type='coh', addLegend=FALSE)
```

---

nyse	<i>Returns of the New York Stock Exchange</i>
------	---

---

**Description**

Returns of the New York Stock Exchange (NYSE) from February 2, 1984 to December 31, 1991.

**Format**

The format is: Time-Series [1:2000] from 1 to 2000: 0.00335 -0.01418 -0.01673 0.00229 -0.01692 ...

**Note**

Various packages have data sets called nyse. Consequently, it may be best to specify this data set as `nyse = astsa : nyse` to avoid conflicts.

**Source**

Most likely from the S+GARCH module - Version 1.1 Release 2: 1998

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

---

oil	<i>Crude oil, WTI spot price FOB</i>
-----	--------------------------------------

---

**Description**

Crude oil, WTI spot price FOB (in dollars per barrel), weekly data from 2000 to mid-2010.

**Format**

The format is: Time-Series [1:545] from 2000 to 2010: 26.2 26.1 26.3 24.9 26.3 ...

**Details**

pairs with the series `gas`

**Source**

Data were obtained from the URL: [www.eia.doe.gov/dnav/pet/pet\\_pri\\_spt\\_s1\\_w.htm](http://www.eia.doe.gov/dnav/pet/pet_pri_spt_s1_w.htm)

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**See Also**

[gas](#)

**Examples**

```
tsplot(cbind(gas,oil), spag=TRUE, col=2*1:2, addLegend=TRUE)
```

---

part

*Particulate levels from the LA pollution study*

---

**Description**

Particulate series corresponding to `cmort` from the LA pollution study.

**Format**

The format is: Time-Series [1:508] from 1970 to 1980: 72.7 49.6 55.7 55.2 66 ...

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**See Also**

[lap](#), [lap.xts](#)

---

PLT

*Platelet Levels*

---

### Description

PLT: Measurements made for 91 days on the three variables, log(white blood count) [WBC], log(platelet) [PLT] and hematocrit [HCT]. Missing data code is 0 (zero).

### Usage

```
data(PLT)
```

### Format

The format is: Time-Series [1:91] from 1 to 91: 4.47 4.33 4.09 4.6 4.41 ...

### Details

See Examples 6.1 and 6.9 for more details.

### Source

Jones, R.H. (1984). Fitting multivariate models to unequally spaced data. In *Time Series Analysis of Irregularly Observed Data*, pp. 158-188. E. Parzen, ed. Lecture Notes in Statistics, 25, New York: Springer-Verlag.

### References

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

### See Also

[blood](#), [HCT](#), [WBC](#)

---

polio

*Poliomyelitis cases in US*

---

### Description

Monthly time series of poliomyelitis cases reported to the U.S. Centers for Disease Control for the years 1970 to 1983, 168 observations.

### Format

The format is: Time-Series [1:168] from 1970 to 1984: 0 1 0 0 1 3 9 2 3 5 ...

### Details

The data were originally modelled by Zeger (1988) "A Regression Model for Time Series of Counts," *Biometrika*, 75, 822-835.

### Source

Data taken from the `gamlss.data` package; see <https://www.gamlss.com/>.

### References

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

### Examples

```
tspplot(polio, type='s')
```

---

polyMul

*Multiplication of Two Polynomials*

---

### Description

Multiplication of two polynomials.

### Usage

```
polyMul(p, q)
```

**Arguments**

p                    coefficients of first polynomial  
q                    coefficients of second polynomial

**Details**

inputs are vectors of coefficients a, b, c, ..., in order of power  $ax^0 + bx^1 + cx^2 + \dots$

**Value**

coefficients of the product in order of power

**Author(s)**

D.S. Stoffer

**Source**

based on code from the polynom package <https://CRAN.R-project.org/package=polynom>

**References**

You can find demonstrations of astsa capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**Examples**

```
a = 1:3 # 1 + 2x + 3x^2
b = 1:2 # 1 + 2x
polyMul(a, b)
# 1 + 4x + 7x^2 + 6x^3
```

---

pre.white

*Cross-Correlation Analysis With Automatic Prewhitening*

---

**Description**

Performs a cross-correlation analysis on two series after prewhitening the first series and filtering the second series accordingly.

**Usage**

```
pre.white(series1, series2, diff = FALSE, max.lag = NULL, main = NULL,
           order.max = NULL, plot = TRUE, ...)
```

**Arguments**

series1, series2	univariate time series
diff	(logical or integer) should the series be differenced prior to the analysis and if more than first order, by how much
max.lag	maximum lag for which to plot the CCF - if NULL, a suitable number is chosen (see details)
main	plot title - if NULL, uses series1 name appended by .w for whitened and series2 name appended by .f for filtered
order.max	maximum order of model to fit (see details)
plot	should the sample CCF be plotted
...	additional graphic arguments

**Details**

The first series is prewhitened by fitting a long AR based on AIC and the second series is filtered appropriately. Then a cross-correlation analysis is performed via `ccf2`. If differencing is specified, both series are differenced the same way prior to the prewhitening. The resulting series are returned invisibly.

The default is no differencing. Differences of order 1 can be set by entering `diff = TRUE` or `diff = 1`. If it is necessary to use higher orders, then enter a positive integer (this is rare).

The maximum lag (`max.lag`) in the CCF graphic defaults (if NULL) to the smaller of 50 and 20% of the sample size.

The maximum order (`order.max`) for fitting the AR via AIC defaults (if NULL) to the minimum of 30 and 15% of the number of observations.

**Value**

Returns the sample CCF graphic using the prewhitened series unless `plot = FALSE`. The prewhitened series are returned invisibly.

**Author(s)**

D.S. Stoffer

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**Examples**

```
pre.white(cmort, part, diff=TRUE, col=4)
```

---

prodn	<i>Monthly Federal Reserve Board Production Index</i>
-------	---

---

**Description**

Monthly Federal Reserve Board Production Index (1948-1978, n = 372 months).

**Usage**

```
data(prodn)
```

**Format**

The format is: Time-Series [1:372] from 1948 to 1979: 40.6 41.1 40.5 40.1 40.4 41.2 39.3 41.6 42.3 43.2 ...

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

---

qinfl	<i>Quarterly Inflation</i>
-------	----------------------------

---

**Description**

Quarterly inflation rate in the Consumer Price Index from 1953-I to 1980-II, n = 110 observations.

**Format**

The format is: Time-Series [1:110] from 1953 to 1980: 1.673 3.173 0.492 -0.327 -0.333 ...

**Details**

pairs with `qintr` (interest rate)

**Source**

Newbold, P. and T. Bos (1985). *Stochastic Parameter Regression Models*. Beverly Hills: Sage.

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**See Also**

[qintr](#)

---

qintr

*Quarterly Interest Rate*

---

**Description**

Quarterly interest rate recorded for Treasury bills from 1953-Ito 1980-II, n = 110 observations.

**Format**

The format is: Time-Series [1:110] from 1953 to 1980: 1.98 2.15 1.96 1.47 1.06 ...

**Details**

pairs with `qinfl` (inflation)

**Source**

Newbold, P. and T. Bos (1985). *Stochastic Parameter Regression Models*. Beverly Hills: Sage.

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**See Also**

[qinfl](#)

---

 QQnorm

*Normal Quantile-Quantile Plot*


---

**Description**

Produces a normal QQ plot with a line of equality and a confidence band (by default) of the input data. This is basically a prettier version of qqnorm from the stats package.

**Usage**

```
QQnorm(xdata, col = c(4, 6), ylab = "Sample Quantiles", xlab = "Theoretical Quantiles",
       main = "Normal Q-Q Plot", ylim = NULL, ci = TRUE, qqld = 1, ...)
```

**Arguments**

xdata	the data. If a matrix, the data are collapsed.
col	vector of 2, first is point color, second is line color (default is blue-4 and magenta-6).
ylab	y-axis label (default is 'Sample Quantiles').
xlab	x-axis label (default is 'Theoretical Quantiles').
main	plot title (default is 'Normal Q-Q Plot')
ylim	limits on y-axis (default is the most beautiful limits ever).
ci	if TRUE (default) draws pointwise 99.99% CIs as a band. If FALSE or 0, no CI is drawn. Alternately, enter a percentage (e.g., either ci=95 or ci=.95 will work).
qqld	line width of the qqline (default is 1).
...	other graphical parameters sent to <a href="#">tsplot</a> .

**Details**

If you want a graphic to check normality of your data in xdata, just enter QQnorm(xdata) and sit back and enjoy the beauty of this script (you may want to wear sunglasses).

For confidence levels, various values are allowed. For example, 95% limits can be obtained as ci=95 or ci=.95, both of which are conventional. However, ci=5, or ci=.05 will also work for 95% intervals (so you can not go below 50%). If you ask for a confidence level of 100% or larger, you will get the default without a warning and maybe you are unconventional.

**Author(s)**

D.S. Stoffer

## References

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

## Examples

```
QQnorm(log(varve))
```

---

rec	<i>Recruitment (number of new fish index)</i>
-----	---

---

## Description

Recruitment (index of the number of new fish) for a period of 453 months ranging over the years 1950-1987. Recruitment is loosely defined as an indicator of new members of a population to the first life stage at which natural mortality stabilizes near adult levels.

## Usage

```
data(rec)
```

## Format

The format is: Time-Series [1:453] from 1950 to 1988: 68.6 68.6 68.6 68.6 68.6 ...

## Details

can pair with `soi` (Southern Oscillation Index)

## Source

Data furnished by Dr. Roy Mendelssohn of the Pacific Fisheries Environmental Laboratory, NOAA (personal communication). Further discussion of the concept of Recruitment may be found here: [derekogle.com/fishR/examples/oldFishRVignettes/StockRecruit.pdf](http://derekogle.com/fishR/examples/oldFishRVignettes/StockRecruit.pdf)

## References

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**See Also**[soi](#)

---

sales	<i>Sales</i>
-------	--------------

---

**Description**

Sales, 150 months; taken from Box and Jenkins (1970).

**Format**

The format is: Time-Series [1:150] from 1 to 150: 200 200 199 199 199 ...

**Details**

This is also the R data set `BJsales`: The sales time series `BJsales` and leading indicator `BJsales.lead` each contain 150 observations. The objects are of class "ts".

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**See Also**[lead](#)

---

salmon	<i>Monthly export price of salmon</i>
--------	---------------------------------------

---

**Description**

Farm Bred Norwegian Salmon, export price, US Dollars per Kilogram

**Format**

The format is: Time-Series [1:166] from September 2003 to June 2017: 2.88 3.16 2.96 3.12 3.23 3.32 3.45 3.61 3.48 3.21 ...

**Source**

<https://www.indexmundi.com/commodities/>

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

---

salt

*Salt Profiles*

---

**Description**

Salt profiles taken over a spatial grid set out on an agricultural field, 64 rows at 17-ft spacing.

**Usage**

```
data(salt)
```

**Format**

The format is: Time-Series [1:64] from 1 to 64: 6 6 6 3 3 3 4 4 4 1.5 ...

**Details**

pairs with `saltemp`, temperature profiles on the same grid

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**See Also**

[saltemp](#)

---

saltemp	<i>Temperature Profiles</i>
---------	-----------------------------

---

**Description**

Temperature profiles over a spatial grid set out on an agricultural field, 64 rows at 17-ft spacing.

**Usage**

```
data(saltemp)
```

**Format**

The format is: Time-Series [1:64] from 1 to 64: 5.98 6.54 6.78 6.34 6.96 6.51 6.72 7.44 7.74 6.85  
...

**Details**

pairs with `salt`, salt profiles on the same grid

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**See Also**

[salt](#)

---

sarima	<i>Fit ARIMA Models</i>
--------	-------------------------

---

**Description**

Fits ARIMA models (with diagnostics) in a short command. It can also be used to perform regression with autocorrelated errors.

**Usage**

```
sarima(xdata, p, d, q, P = 0, D = 0, Q = 0, S = -1,  
       details = TRUE, xreg = NULL, Model = TRUE,  
       fixed = NULL, tol = sqrt(.Machine$double.eps),  
       no.constant = FALSE, col, fitdf, ...)
```

**Arguments**

xdata	univariate time series
p	AR order
d	difference order
q	MA order
P	SAR order; use only for seasonal models
D	seasonal difference; use only for seasonal models
Q	SMA order; use only for seasonal models
S	seasonal period; use only for seasonal models
details	if FALSE, turns off the diagnostic plot and the output from the nonlinear optimization routine, which is <code>optim</code> . The default is TRUE.
xreg	Optionally, a vector or matrix of external regressors, which must have the same number of rows as xdata.
Model	if TRUE (default), the model orders are printed on the diagnostic plot.
fixed	optional numeric vector of the same length as the total number of parameters. If supplied, only parameters corresponding to NA entries will be estimated.
tol	controls the relative tolerance ( <code>reltol</code> in <code>optim</code> ) used to assess convergence. The default is <code>sqrt(.Machine\$double.eps)</code> , the R default.
no.constant	controls whether or not <code>sarima</code> includes a constant in the model. In particular, if there is no differencing ( $d = 0$ and $D = 0$ ) you get the mean estimate. If there is differencing of order one (either $d = 1$ or $D = 1$ , but not both), a constant term is included in the model. These two conditions may be overridden (i.e., no constant will be included in the model) by setting this to TRUE; e.g., <code>sarima(x, 1, 1, 0, no.constant=TRUE)</code> . Otherwise, no constant or mean term is included in the model. If regressors are included (via <code>xreg</code> ), this is ignored.
col	color of diagnostic plots; default is 1 (black)
fitdf	number of degrees of freedom to be subtracted for the Ljung-Box test if xdata is a series of residuals and all orders are zero (see details). Does not have to be specified if it is zero (0).
...	additional graphical arguments

**Details**

If your time series is in `x` and you want to fit an ARIMA( $p,d,q$ ) model to the data, the basic call is `sarima(x,p,d,q)`. As of version 2.3, the orders do not have to be specified if they are zero. For example, `sarima(x, p=1)` is the same as `sarima(x, 1, 0, 0)`.

To fit a seasonal ARIMA model, the basic call is `sarima(x, p, d, q, P, D, Q, S)`. For example, `sarima(x, 2, 1, 0, 0, 1, 1, 12)` will fit a seasonal ARIMA( $2, 1, 0$ )  $\times$  ( $0, 1, 1$ )<sub>12</sub> model to the series in `x`. The orders do not have to be specified if they are zero; e.g., `sarima(x, d=1, q=1, D=1, Q=1, S=4)` works.

The results are the parameter estimates, standard errors, AIC, AICc, BIC and diagnostics. The difference between the information criteria given by `sarima()` and `arima()` is that they differ by a scaling factor of the effective sample size.

The script may be used for a residual analysis by running it without specifying any orders. In this case, it may be necessary to specify `fitdf` to get the correct degrees of freedom such as in fitting a state space model.

**Value**

A t-table, the estimated noise variance, and AIC, AICc, BIC are printed. The following are returned invisibly as a list:

```
fit          [[1]] an object of class Arima with more information than you need
sigma2       [[2]] the estimate of the noise variance
degrees_of_freedom
              [[3]] error degrees of freedom
t.table      [[4]] a little t-table with two-sided p-values
ICs          [[5]] AIC - AICc - BIC
```

**Missing Data**

Yes it's ok if input as NA and the observations are vector or ts objects (meaning equally spaced).

**Source**

This is an enhancement of arima from the stats package.

**References**

You can find demonstrations of astsa capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**See Also**

[sarima.for](#), [sarima.sim](#)

**Examples**

```
# easy to use
sarima(rec, 2,0,0) # data, p, d, and q

# redux - minimal output
sarima(rec, p=2, details=FALSE)

# fun for the whole family
dog = sarima(log(AirPassengers), 0,1,1, 0,1,1,12, details=FALSE)
# dog[[1]] has most of the results ...
tsplot(resid(dog[[1]]))

# fixed parameters
x = sarima.sim( ar=c(0,-.9), n=200 ) + 50
sarima(x, p=2, fixed=c(0,NA,NA)) # phi1 fixed, phi2 and mean free
```

```
# regression with autocorrelated errors
sarima(log(cpg), p=1, xreg=time(cpg))

# missing data (color, gris-gris, and pch, added for fun)
sarima(ar1miss, p=1, col=4, gg=TRUE, pch=19)
```

---

 sarima.for

*ARIMA Forecasting*


---

### Description

ARIMA forecasting.

### Usage

```
sarima.for(xdata, n.ahead, p, d, q, P=0, D=0, Q=0, S=-1, tol = sqrt(.Machine$double.eps),
  no.constant = FALSE, plot = TRUE, plot.all = FALSE, ylab = NULL, xreg = NULL,
  newxreg = NULL, fixed = NULL, pcol = 2, pch = 1, ...)
```

### Arguments

xdata	univariate time series
n.ahead	forecast horizon (number of periods)
p	AR order
d	difference order
q	MA order
P	SAR order; use only for seasonal models
D	seasonal difference; use only for seasonal models
Q	SMA order; use only for seasonal models
S	seasonal period; use only for seasonal models
tol	controls the relative tolerance (reltol) used to assess convergence. The default is <code>sqrt(.Machine\$double.eps)</code> , the R default.
no.constant	controls whether or not a constant is included in the model. If <code>no.constant=TRUE</code> , no constant is included in the model. See <a href="#">sarima</a> for more details.
plot	if TRUE (default) the data (or some of it) and the forecasts and bounds are plotted
plot.all	if TRUE, all the data are plotted in the graphic; otherwise, only the last 100 observations are plotted in the graphic.
ylab	if NULL (default), the y-axis label is the name of the series.
xreg	Optionally, a vector or matrix of external regressors, which must have the same number of rows as the series. If this is used, <code>newxreg</code> MUST be specified.

<code>newxreg</code>	New values of <code>xreg</code> to be used for prediction. Must have at least <code>n.ahead</code> rows.
<code>fixed</code>	optional numeric vector of the same length as the total number of parameters. If supplied, only parameters corresponding to NA entries will be estimated.
<code>pcol</code>	color of the predictions in the graphic.
<code>pch</code>	plot character for the graphic. If <code>plot.all=TRUE</code> , this only applies to the forecasts.
<code>...</code>	additional graphical arguments

### Details

For example, `sarima.for(x, 5, 1, 0, 1)` or `sarima.for(x, 5, p=1, q=1)` will forecast five time points ahead for an ARMA(1,1) fit to `x`. The output prints the forecasts and the standard errors of the forecasts, and supplies a graphic of the forecast with +/- 1 and 2 prediction error bounds.

### Value

<code>pred</code>	the forecasts
<code>se</code>	the prediction (standard) errors

### Missing Data

Yes it's ok if input as NA and the observations are vector or `ts` objects (meaning equally spaced). In this case, the graphic includes a line with points. Otherwise, lone observations would not be visible.

### Note

If `plot.all=TRUE`, the data are displayed as a line only unless there are missing observations; see the Missing Data section. Points (and more) can be added to the graphic as long as the device stays open. For example:

```
sarima.for(gtemp_land, 10, d=1, q=1, plot.all=TRUE, pch=19)
points(gtemp_land, pch=20, col=4)
abline(v=2024, col=6, lty=5)
text(2000, 2.2, "it's getting hot in here", font=2, col=6, srt=45)
```

### References

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

### See Also

[sarima](#)

**Examples**

```

sarima.for(gtemp_both, n.ahead=6, d=1, q=1, col=6, pcol=4, gg=TRUE)

# with regressors
nummy = length(soi)
n.ahead = 24
nureg = time(soi)[nummy] + seq(1,n.ahead)/12
sarima.for(soi, n.ahead, 2,0,0, 2,0,0,12, xreg=time(soi), newxreg=nureg)

# missing data
sarima.for(ar1miss, n.ahead=5, p=1, pch=19)

```

---

sarima.sim

*ARIMA Simulation*


---

**Description**

Simulate data from (seasonal) ARIMA models.

**Usage**

```

sarima.sim(ar = NULL, d = 0, ma = NULL, sar = NULL, D = 0, sma = NULL, S = NULL,
           n = 500, rand.gen = rnorm, innov = NULL, burnin = NA, t0 = 0, red.tol=.1, ...)

```

**Arguments**

ar	coefficients of AR component (does not have to be specified)
d	order of regular difference (does not have to be specified)
ma	coefficients of MA component (does not have to be specified)
sar	coefficients of SAR component (does not have to be specified)
D	order of seasonal difference (does not have to be specified)
sma	coefficients of SMA component (does not have to be specified)
S	seasonal period (does not have to be specified)
n	desired sample size (defaults to 500)
rand.gen	optional; a function to generate the innovations (defaults to normal)
innov	an optional times series of innovations. If not provided, rand.gen is used.
burnin	length of burn-in (a non-negative integer). If NA, or not a positive integer, or a vector, a reasonable value is selected.
t0	start time (defaults to 0)
red.tol	tolerance for reporting parameter redundancy (default is .1 - set to 0 to avoid check)
...	additional arguments applied to the innovations. For rand.gen, the standard deviation of the innovations generated by rnorm can be specified by sd or the mean by mean (see details and examples). In addition, rand.gen may be overridden using a preset sequence of innovations specifying innov (see details and examples).

## Details

Will generate a time series of length  $n$  from the specified SARIMA model using simplified input.

The use of the term `mean` under the `'...'` argument refers to the generation of normal innovations. For example, `sarima.sim(ar=.9, mean=5)` will generate data using  $N(5,1)$  or  $5+N(0,1)$  innovations, so that the constant in the model is 5 and the mean of the AR model is  $5/(1-.9) = 50$ . In `sarima.sim(ma=.9, mean=5)`, however, the model mean is 5 (the constant). Also, a random walk with drift = .1 can be generated by `sarima.sim(d=1, mean=.1, burnin=0)`, which is equivalent to `cumsum(rnorm(500, mean=.1))`. The same story goes if `sd` is specified; i.e., it's applied to the innovations. Because anything specified in `...` refers to the innovations, a simpler way to generate a non-zero mean is to add the value outside the call; see the examples.

If `innov` is used to input the innovations and override `rand.gen`, be sure that `length(innov)` is at least  $n + \text{burnin}$ . If the criterion is not met, the script will return less than the desired number of values and a warning will be given.

## Value

A time series of length  $n$  from the specified SARIMA model with the specified frequency if the model is seasonal and start time `t0`.

## Note

The model autoregressive polynomial ('AR side' = AR x SAR) is checked for causality and the model moving average polynomial ('MA side' = MA x SMA) is checked invertibility. The script stops and reports an error at the first violation of causality or invertibility; i.e., it will not report multiple errors.

Overparameterization is also checked. To evaluate parameter redundancy, the inverse roots of the (S)AR and (S)MA polynomials are examined for closeness with `red.tol` determining closeness. This can be shut off by setting `red.tol=0`.

## Author(s)

D.S. Stoffer

## References

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

## Examples

```
## AR(2) with mean 50 [n = 500 is default]
y = sarima.sim(ar=c(1.5,-.75)) + 50
tsplot(y)
```

```

## ARIMA(0,1,1) with drift ['mean' refers to the innovations]
tsplot(sarima.sim(ma=-.8, d=1, mean=.25))

## Overparameterized White Noise
tsplot(sarima.sim(ar=.9, ma=-.9))

## SAR(1) example from text
set.seed(10101010)
x = sarima.sim(sar=.95, S=12, n=37) + 5
tsplot(x, type='c')
points(x, pch=Months, cex=1.5, font=2, col=1:12)

## SARIMA(0,1,1)x(0,1,1)_12 - B&J's favorite
set.seed(101010)
tsplot(sarima.sim(d=1, ma=-.4, D=1, sma=-.6, S=12, n=120))

## infinite variance t-errors
tsplot(sarima.sim(ar=.9, rand.gen=function(n, ...) rt(n, df=2) ))

## use your own innovations
dog = rexp(150, rate=.5)*sign(runif(150,-1,1))
tsplot(sarima.sim(n=100, ar=.99, innov=dog, burnin=50))

## generate seasonal data but no P, D or Q - you will receive
## a message to make sure that you wanted to do this on purpose:
tsplot(sarima.sim(ar=c(1.5,-.75), n=144, S=12), ylab='doggy', xaxt='n')
mtext(seq(0,144,12), side=1, line=.5, at=0:12)

```

---

scatter.hist

*Scatterplot with Marginal Histograms*


---

## Description

Draws a scatterplot with histograms in the margins.

## Usage

```

scatter.hist(x, y, xlab = NULL, ylab = NULL, title = NULL, pt.size = 1,
             hist.col = gray(0.82), pt.col = gray(0.1, 0.25), pch = 19,
             reset.par = TRUE, ...)

```

## Arguments

x	vector of x-values
y	corresponding vector of y-values
xlab	x-axis label (defaults to name of x)
ylab	y-axis label (defaults to name of y)

title	plot title (optional)
pt.size	size of points in scatterplot
hist.col	color for histograms
pt.col	color of points in scatterplot
pch	scatterplot point character
reset.par	reset graphics - default is TRUE; set to FALSE to add on to scatterplot
...	additional graphical parameters sent to Grid

**Author(s)**

D.S. Stoffer

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**Examples**

```
scatter.hist(temp, cmort, hist.col=astsa.col(5,.4), pt.col=5, pt.size=1.5, reset=FALSE)
lines(lowess(temp, cmort), col=6)
```

```
scatter.hist(diff(log(econ5[, 'gnp'])), diff(log(econ5[, 'unemp'])), nxm=5, pt.size=2,
  col=astsa.col(6,.1), pt.col=astsa.col(4, alpha=.6, wheel=TRUE, num=25), hist='green3')
```

---

SigExtract

*Signal Extraction And Optimal Filtering*

---

**Description**

Performs signal extraction and optimal filtering as discussed in Chapter 4.

**Usage**

```
SigExtract(series, L = c(3, 3), M = 50, max.freq = 0.05, col = 4)
```

**Arguments**

series	univariate time series to be filtered
L	degree of smoothing (may be a vector); see spans in spec.pgram for more details
M	number of terms used in the lagged regression approximation
max.freq	truncation frequency, which must be larger than 1/M
col	color of the main graphs

**Details**

The basic function of the script, and the default setting, is to remove frequencies above 1/20 (and, in particular, the seasonal frequency of 1 cycle every 12 time points). The sampling frequency of the time series is set to unity prior to the analysis.

**Value**

Returned invisibly as a list

series.filt	[[1]] the filtered series
filter	[[2]] the filter

Also prints the rounded filter coefficients and returns plots of (1) the original and filtered series, (2) the estimated spectra of each series, (3) the filter coefficients and the desired and attained frequency response function.

**Note**

The script is based on code that was contributed by Professor Doug Wiens, Department of Mathematical and Statistical Sciences, University of Alberta.

**Author(s)**

D.S. Stoffer

**References**

You can find demonstrations of astsa capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**Examples**

```
## Not run:

SigExtract(detrend(soi)) -> u
tspplot(cbind(soi.filtered=u[[1]], detrend(soi)), col=4*1:2, lwd=2:1, spag=TRUE, addLegend=TRUE)

## End(Not run)
```

---

sleep1

*Sleep State and Movement Data - Group 1*


---

**Description**

Sleep-state and number of movements of infants taken from a study on the effects of prenatal exposure to alcohol. This is Group 1 where the mothers did not drink alcohol during pregnancy.

**Format**

List of 12 (by subjects) 'data.frame': 120 obs. of 3 variables: .. min : int [1:120] minute (1 to 120)  
 .. state: int [1:120] sleep state 1 to 6 with NA missing (see details) .. mvmnt: int [1:120] number of  
 movements

**Details**

Per minute sleep state, for approximately 120 minutes, is categorized into one of six possible states, non-REM: NR1 [1] to NR4 [4], and REM [5], or AWAKE [6]. NA means no state is recorded for that minute (if there, it occurs at end of the session). Group 1 (this group) is from mothers who abstained from drinking during pregnancy. In addition, the number of movements per minute are listed.

**Source**

Stoffer, D. S., Scher, M. S., Richardson, G. A., Day, N. L., Coble, P. A. (1988). A Walsh-Fourier Analysis of the Effects of Moderate Maternal Alcohol Consumption on Neonatal Sleep-State Cycling. *Journal of the American Statistical Association*, 83(404), 954-963. <https://doi.org/10.2307/2290119>

Stoffer, D. S. (1990). Multivariate Walsh-Fourier Analysis. *Journal of Time Series Analysis*, 11(1), 57-73. <https://doi.org/10.1111/j.1467-9892.1990.tb00042.x>

**References**

You can find demonstrations of astsa capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**See Also**[sleep2](#)**Examples**

```
# spectral analysis
x = dna2vector(sleep1[[1]]$state, alphabet=c('1','2','3','4','5')) # never awake
specenv(na.omit(x), spans=c(3,3), col=5) # not all babies make it to 120 minutes
abline(v=1/60, lty=2, col=8)
mtext(side=1, '1/60', at=1/60, cex=.75)
```

sleep2

*Sleep State and Movement Data - Group 2***Description**

Sleep-state and number of movements of infants taken from a study on the effects of prenatal exposure to alcohol. This is Group 2 where the mothers drank alcohol in moderation during pregnancy.

**Format**

List of 12 (by subjects) :'data.frame': 120 obs. of 3 variables: .. min : int [1:120] minute (1 to 120)  
 .. state: int [1:120] sleep state 1 to 6 with NA missing (see details) .. mvmnt: int [1:120] number of  
 movements

**Details**

Per minute sleep state, for approximately 120 minutes, is categorized into one of six possible states, non-REM: NR1 [1] to NR4 [4], and REM [5], or AWAKE [6]. NA means no state is recorded for that minute (if there, it occurs at end of the session). Group 2 (this group) is from mothers who drank alcohol in moderation during pregnancy. In addition, the number of movements per minute are listed.

**Source**

Stoffer, D. S., Scher, M. S., Richardson, G. A., Day, N. L., Coble, P. A. (1988). A Walsh-Fourier Analysis of the Effects of Moderate Maternal Alcohol Consumption on Neonatal Sleep-State Cycling. *Journal of the American Statistical Association*, 83(404), 954-963. <https://doi.org/10.2307/2290119>

Stoffer, D. S. (1990). Multivariate Walsh-Fourier Analysis. *Journal of Time Series Analysis*, 11(1), 57-73. <https://doi.org/10.1111/j.1467-9892.1990.tb00042.x>

## References

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

## See Also

[sleep1](#)

## Examples

```
tspplot(sleep2[[1]][,1], sleep2[[1]][,2:3], type='s', col=2*2:3, xlab='minute')
title('Baby Goo from Group 2', outer=TRUE, line=-1, cex.main=1)
```

---

so2

*SO2 levels from the LA pollution study*

---

## Description

Sulfur dioxide levels from the LA pollution study

## Format

The format is: Time-Series [1:508] from 1970 to 1980: 3.37 2.59 3.29 3.04 3.39 2.57 2.35 3.38 1.5 2.56 ...

## References

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

## See Also

[lap](#)

soi

*Southern Oscillation Index*

---

**Description**

Southern Oscillation Index (SOI) for a period of 453 months ranging over the years 1950-1987.

**Format**

The format is: Time-Series [1:453] from 1950 to 1988: 0.377 0.246 0.311 0.104 -0.016 0.235 0.137 0.191 -0.016 0.29 ...

**Details**

pairs with rec (Recruitment)

**Source**

Data furnished by Dr. Roy Mendelsohn of the Pacific Fisheries Environmental Laboratory, NOAA (personal communication).

**References**

You can find demonstrations of astsa capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**See Also**

[rec](#), [ENSO](#), [MEI](#), [MEI2](#)

---

soiltemp

*Spatial Grid of Surface Soil Temperatures*

---

**Description**

A 64 by 36 matrix of surface soil temperatures (°C).

**Format**

The format is: num [1:64, 1:36] 6.7 8.9 5 6.6 6.1 7 6.5 8.2 6.7 6.6 ...

**Source**

Bazza, M., Shumway, R. H., & Nielsen, D. R. (1988). Two-dimensional spectral analysis of soil surface temperature. *Hilgardia: A Journal of Agricultural Science*, 56, 1-28.

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**Examples**

```
par(mar = rep(1,4))
persp(1:64, 1:36, soiltemp, phi=25, theta=25, scale=FALSE, expand=4,
      col='lightblue', border=4, ticktype="detailed", xlab="rows",
      ylab="columns", zlab="temperature")
```

---

sp500.gr

*Returns of the S&P 500*

---

**Description**

Daily growth rate of the S&P 500 from 2001 though 2011.

**Format**

The format is: Time Series; Start = c(2001, 2); End = c(2011, 209); Frequency = 252

**Source**

Douc, Moulines, & Stoffer (2014). *Nonlinear Time Series: Theory, Methods and Applications with R Examples*. CRC Press. doi:10.1201/b16331.

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

---

`sp500w`*Weekly Growth Rate of the Standard and Poor's 500*

---

**Description**

Weekly closing returns of the SP 500 from 2003 to September, 2012.

**Format**

An 'xts' object on 2003-01-03 to 2012-09-28; Indexed by objects of class: [Date] TZ: UTC

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**Examples**

```
# if 'xts' is not loaded, you can do this
par(mfrow=2:1)
tsplot(sp500w, col=4, lwd=2)           # no dates
tsplot(timex(sp500w), sp500w, col=4, lwd=2) # dates
```

---

`spec.ic`*Estimate Spectral Density of a Time Series from AR Fit*

---

**Description**

Fits an AR model to data and computes (and by default plots) the spectral density of the fitted model based on AIC (default) or BIC.

**Usage**

```
spec.ic(xdata, BIC=FALSE, order.max=NULL, main=NULL, plot=TRUE, detrend=TRUE,
        lowess=FALSE, method=NULL, cex.main=NULL, xlab=NULL, ...)
```

**Arguments**

xdata	a univariate time series.
BIC	if TRUE, fit is based on BIC. If FALSE (default), fit is based on AIC.
order.max	maximum order of model to fit. Defaults (if NULL) to the minimum of 100 and 10% of the number of observations.
main	plot title. Defaults to name of series, method and chosen order.
plot	if TRUE (default) produces a graphic of the estimated AR spectrum.
detrend	if TRUE (default), detrends the data first. If FALSE, the series is demeaned.
lowess	if TRUE, detrends using lowess. Default is FALSE.
method	method of estimation - a character string specifying the method to fit the model chosen from the following: "yule-walker", "burg", "ols", "mle", "yw". Defaults to "yule-walker".
cex.main	magnification for main title; default is 1.
xlab	label for frequency axis; if NULL (default), a totally awesome label is generated for your viewing pleasure.
...	additional graphical arguments.

**Details**

Uses ar to fit the best AR model based on pseudo AIC or BIC. Using method='mle' will be slow. The minimum centered AIC and BIC values and the spectral and frequency ordinates are returned silently.

**Value**

[[1]]	Matrix with columns: ORDER, AIC, BIC
[[2]]	Matrix with columns: freq, spec

**Author(s)**

D.S. Stoffer

**References**

You can find demonstrations of astsa capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**See Also**

[ar](#), [spec.ar](#)

**Examples**

```
# AIC
dog <- spec.ic(soi) # put results in dog

# plot AIC and BIC
tsplot(dog[[1]][,2:3], col=2*1:2, type='o', pch=1:2, xlab='order',
       spag=TRUE, addLegend=TRUE)
```

---

specenv

*Spectral Envelope*


---

**Description**

Computes the spectral envelope of categorical-valued or real-valued time series.

**Usage**

```
specenv(xdata, section = NULL, spans = NULL, kernel = NULL, taper = 0,
       significance = 1e-04, plot = TRUE, ylim = NULL, real = FALSE, ...)
```

**Arguments**

xdata	For categorical-valued sequences, a matrix with rows that are indicators of the categories represented by the columns, possibly a sequence converted using <a href="#">dna2vector</a> . For real-valued sequences, a matrix with at least two columns that are various transformations of the data.
section	of the form <code>start:end</code> where <code>start &lt; end</code> are positive integers; specifies the section used in the analysis - default is the entire sequence.
spans	specify smoothing used in <code>mvspec</code> .
kernel	specify kernel to be used in <code>mvspec</code> .
taper	specify amount of tapering to be used in <code>mvspec</code> .
significance	significance threshold exhibited in plot - default is <code>.0001</code> ; set to <code>NA</code> to cancel
plot	if <code>TRUE</code> (default) a graphic of the spectral envelope is produced
ylim	limits of the spectral envelope axis; if <code>NULL</code> (default), a suitable range is calculated.
real	<code>FALSE</code> (default) for categorical-valued sequences and <code>TRUE</code> for real-valued sequences.
...	other graphical parameters.

**Details**

Calculates the spectral envelope for categorical-valued series as discussed in [https://www.stat.pitt.edu/stoffer/dss\\_files/spenv.pdf](https://www.stat.pitt.edu/stoffer/dss_files/spenv.pdf) and summarized in

<https://doi.org/10.1214/ss/1009212816>.

Alternately, calculates the spectral envelope for real-valued series as discussed in [https://doi.org/10.1016/S0378-3758\(96\)00044-4](https://doi.org/10.1016/S0378-3758(96)00044-4).

These concepts are also presented (with examples) in Section 7.9 (Chapter 7) of Time Series Analysis and Its Applications: With R Examples: <https://www.stat.pitt.edu/stoffer/tsa4/>.

For categorical-valued series, the input `xdata` must be a matrix of indicators which is perhaps a sequence preprocessed using `dna2vector`.

For real-valued series, the input `xdata` should be a matrix whose columns are various transformations of the univariate series.

The script does not detrend the data prior to estimating spectra. If this is an issue, then detrend the data prior to using this script.

**Value**

By default, will produce a graph of the spectral envelope and an approximate significance threshold. A matrix containing: frequency, spectral envelope ordinates, and (1) the scalings of the categories in the order of the categories in the alphabet or (2) the coefficients of the transformations, is returned invisibly.

**Author(s)**

D.S. Stoffer

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**See Also**

[dna2vector](#)

**Examples**

```
# a DNA sequence
data = bnrflbv
xdata = dna2vector(data)
u = specenv(xdata, section=1:1000, spans=c(7,7))
head(u) # scalings are for A, C, G, and last one T=0 always
```

```

# a real-valued series
x = astsa::nyse
xdata = cbind(x, abs(x), x^2)
u = specenv(xdata, real=TRUE, spans=c(3,3))
# plot optimal transform at freq = .001
beta = u[2, 3:5]
( b = beta/beta[2] ) # makes abs(x) coef=1
gopt = function(z) { b[1]*z + b[2]*abs(z) + b[3]*z^2 }
gabs = function(z) { abs(z) }
z = -20:20/100
tsplot(z, cbind(gopt(z), gabs(z)), spag=TRUE, col=5:6, addLegend=TRUE, lwd=2,
        xlab='return', legend = c('optimal','absolute value' ), gg=TRUE)
title('transformations')

```

---

speech

*Speech Recording*

---

### Description

A small .1 second (1000 points) sample of recorded speech for the phrase "aaa...hhh".

### Format

The format is: Time-Series [1:1020] from 1 to 1020: 1814 1556 1442 1416 1352 ...

### References

You can find demonstrations of astsa capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

---

ssm

*State Space Model*

---

### Description

Fits a simple univariate state space model to data. The parameters are estimated (the state regression parameter may be fixed). State predictions, filters, and smoothers and corresponding error variances are evaluated at the estimates. The sample size must be at least 20.

**Usage**

```
ssm(y, A, phi, alpha, sigw, sigv, fixphi = FALSE)
```

**Arguments**

y	data
A	measurement value (fixed constant)
phi	initial value of phi, may be fixed
alpha	initial value for alpha
sigw	initial value for sigma[w]
sigv	initial value for sigma[v]
fixphi	if TRUE, the phi parameter is fixed

**Details**

The script works for a specific univariate state space model,

$$x_t = \alpha + \phi x_{t-1} + w_t \quad \text{and} \quad y_t = Ax_t + v_t.$$

The initial state conditions use a default calculation and cannot be specified. The parameter estimates are printed and the script returns the state predictors and smoothers. The regression parameter  $\phi$  may be fixed.

**Value**

At the MLEs, these are returned invisibly:

Xp	time series - state prediction, $x_t^{t-1}$
Pp	corresponding MSPEs, $P_t^{t-1}$
Xf	time series - state filter, $x_t^t$
Pf	corresponding MSEs, $P_t^t$
Xs	time series - state smoother, $x_t^n$
Ps	corresponding MSEs, $P_t^n$

**Author(s)**

D.S. Stoffer

**References**

You can find demonstrations of astsa capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**Examples**

```
## Not run:

u = ssm(gtemp_land, A=1, alpha=.01, phi=1, sigw=.05, sigv=.15, fixphi=TRUE)
tsplot(gtemp_land, type='o', col=4)
lines(u$Xs, col=6, lwd=2)

## End(Not run)
```

---

star

*Variable Star*


---

**Description**

The magnitude of a star taken at midnight for 600 consecutive days.

**Format**

The format is: Time-Series [1:600] from 1 to 600: 25 28 31 32 33 33 32 ...

**Source**

The data are taken from the classic text, *The Calculus of Observations, a Treatise on Numerical Mathematics*, by E.T. Whittaker and G. Robinson, (1923, Blackie and Son, Ltd.).

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

---

stoch.reg

*Frequency Domain Stochastic Regression*


---

**Description**

Performs frequency domain stochastic regression discussed in Chapter 7.

**Usage**

```
stoch.reg(xdata, cols.full, cols.red=NULL, alpha, L, M, plot.which, col.resp=NULL, ...)
```

**Arguments**

xdata	data matrix with the last column being the response variable
cols.full	specify columns of data matrix that are in the full model
cols.red	specify columns of data matrix that are in the reduced model (use NULL if there are no inputs in the reduced model)
alpha	test size; number between 0 and 1
L	odd integer specifying degree of smoothing
M	number (integer) of points in the discretization of the integral
plot.which	coh or F.stat, to plot either the squared-coherencies or the F-statistics, respectively
col.resp	specify column of the response variable if it is not the last column of the data matrix
...	additional graphic arguments

**Details**

This function computes the spectral matrix, F statistics and coherences, and plots them. Returned as well are the coefficients in the impulse response function.

Enter, as the argument to this function, the full data matrix, and then the labels of the columns of input series in the "full" and "reduced" regression models - enter NULL if there are no inputs under the reduced model.

If the response variable is the LAST column of the data matrix, it need not be specified. Otherwise specify which column holds the responses as col.resp.

Other inputs are alpha (test size), L (smoothing), M (number of points in the discretization of the integral) and plot.which = "coh" or "F", to plot either the coherences or the F statistics.

**Value**

power.full	spectrum under the full model
power.red	spectrum under the reduced model
Betahat	regression parameter estimates
eF	pointwise (by frequency) F-tests
coh	coherency

**Note**

See Example 7.1 of the text. The script is based on code that was contributed by Professor Doug Wiens, Department of Mathematical and Statistical Sciences, University of Alberta.

**Author(s)**

D.S. Stoffer

## References

You can find demonstrations of astsa capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

---

sunspotz

*Biannual Sunspot Numbers*

---

## Description

Biannual smoothed (12-month moving average) number of sunspots from June 1749 to December 1978; n = 459.

## Format

The format is: Time Series: Start = c(1749, 1) End = c(1978, 1) Frequency = 2

## Note

The "z" on the end of sunspotz is to distinguish this series from the ones included with R (see [sunspots](#)).

## References

You can find demonstrations of astsa capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

SV.mcmc

*Fit Bayesian Stochastic Volatility Model***Description**

Fits a stochastic volatility model to a univariate time series of returns.

**Usage**

```
SV.mcmc(y, ncmc = 1000, burnin = 100, init = NULL, hyper = NULL, tuning = NULL,
        sigma_MH = NULL, npart = NULL, mcmseed = NULL)
```

**Arguments**

y	single time series of returns
ncmc	number of iterations for the MCMC procedure
burnin	number of iterations to discard for the MCMC procedure
init	initial values of (phi, sigma, beta) - default is c(0.9, 0.5, .1)
hyper	hyperparameters for bivariate normal distribution of (phi, sigma); user inputs (mu_phi, mu_q, sigma_phi, sigma_q, rho) - default is c(0.9, 0.5, 0.075, 0.3, -0.25)
tuning	tuning parameter - default is .03
sigma_MH	covariance matrix used for random walk Metropolis; it will be scaled by tuning in the script - default is matrix(c(1, -.25, -.25, 1), nrow=2, ncol=2)
npart	number of particles used in particle filter - default is 10
mcmseed	seed for mcmc - default is 90210

**Details**

The log-volatility process is  $x_t$  and the returns are  $y_t$ . The SV model is

$$x_t = \phi x_{t-1} + \sigma w_t \quad y_t = \beta \exp\left\{\frac{1}{2}x_t\right\}\epsilon_t$$

where  $w_t$  and  $\epsilon_t$  are independent standard normal white noise.

The model is fit using a technique described in the paper listed below (in the Source section) where the state parameters  $(\phi, \sigma)$  are sampled simultaneously with a bivariate normal prior specified in the arguments `init` and `hyper`.

Two graphics are returned: (1) the three parameter traces with the posterior mean highlighted, their ACFs [with effective sample sizes (ESS)], and their histograms with the .025, .5, and .975 quantiles displayed, and (2) the log-volatility posterior mean along with corresponding .95 credible intervals.

**Value**

Returned invisibly as a list:

phi	[[1]] vector of sampled state AR parameter
sigma	[[2]] vector of sampled state error std deviation
beta	[[3]] vector of sampled observation error scale
log.vol	[[4]] matrix of sampled log-volatility
options	[[5]] values of the input arguments

**Note**

Except for the data, all the other inputs have defaults. The time to run and the acceptance rate are returned at the end of the analysis. The acceptance rate should be around 30% and this is easily adjusted using the tuning parameter.

**Author(s)**

D.S. Stoffer

**Source**

Gong & Stoffer (2021). A note on efficient fitting of stochastic volatility models. *Journal of Time Series Analysis*, 42(2), 186-200. <https://github.com/nickpoison/Stochastic-Volatility-Models>

**References**

You can find demonstrations of astsa capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**See Also**

[SV.mle](#)

**Examples**

```
## Not run:
#-- A minimal example --##
myrun <- SV.mcmc(sp500w) # results in object myrun - don't forget it

str(myrun) # an easy way to see the default input options

## End(Not run)
```

SV.mle

*Stochastic Volatility Model with Feedback via MLE***Description**

Fits a stochastic volatility model with feedback (optional) to a univariate time series of returns via quasi-MLE.

**Usage**

```
SV.mle(returns, gamma = 0, phi = 0.95, sQ = 0.1, alpha = NULL, sR0 = 1, mu1 = -3, sR1 = 2,
       rho = NULL, feedback = FALSE)
```

**Arguments**

returns	single time series of returns
gamma	feedback coefficient - included if feedback=TRUE (does not have to be specified)
phi	initial value of the log-volatility AR parameter (does not have to be specified)
sQ	initial value of the standard deviation of log-volatility noise (does not have to be specified)
alpha	initial value of the log-returns <sup>2</sup> constant parameter (does not have to be specified)
sR0	initial value of the log-returns <sup>2</sup> normal mixture standard deviation parameter (component 0 - does not have to be specified)
mu1	initial value of the log-returns <sup>2</sup> normal mixture mean parameter (component 1 - does not have to be specified)
sR1	initial value of the log-returns <sup>2</sup> normal mixture standard deviation parameter (component 1 - does not have to be specified)
rho	correlation between the state noise and observation noise (so called "leverage"). If feedback=TRUE this will be included if given a proper numerical value; if NULL (default) it is not included because it is often not significant when the feedback coefficient is included.
feedback	if TRUE feedback is included in the model; default is FALSE.

**Details**

The returns are  $r_t$  (input this). The log-volatility process is  $x_t$  and  $y_t = \log r_t^2$ .

If feedback=TRUE, the model is

$$x_{t+1} = \gamma r_t + \phi x_t + \sigma w_t \quad y_t = \alpha + x_t + \eta_t$$

where  $w_t$  is standard normal noise. The observation error  $\eta_t$  is a mixture of two normals,  $N(0, \sigma_0^2)$  and  $N(\mu_1, \sigma_1^2)$ . The state and observation noise can be correlated if  $\rho$  is given a value between -1 and 1.

If feedback=FALSE,  $\gamma$  and  $\rho$  are not included in the model.

**Value**

Returned invisibly:

PredLogVol	one-step-ahead predicted log-volatility
RMSPE	corresponding root MSPE
Coefficients	table of estimates and estimated standard errors

In addition to the one step ahead predicted log-volatility, corresponding root MSPE, and table of estimates returned invisibly, the estimates and SEs are printed and a graph of (1) the data with the predicted log-volatility, and (2) the normal mixture are displayed in one graphic.

**Author(s)**

D.S. Stoffer

**References**

You can find demonstrations of astsa capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**See Also**

[SV.mcmc](#)

**Examples**

```
## Not run:

SV.mle(sp500.gr, feedback=TRUE)

SV.mle(nyse)

## End(Not run)
```

---

tempr

*Temperatures from the LA pollution study*

---

**Description**

Temperature series corresponding to cmort from the LA pollution study.

**Format**

The format is: Time-Series [1:508] from 1970 to 1980: 72.4 67.2 62.9 72.5 74.2 ...

**References**

You can find demonstrations of astsa capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**See Also**

[lap](#), [lap.xts](#)

---

 test.linear

---

*Test Linearity of a Time Series via Normalized Bispectrum*


---

**Description**

Produces a plot of the tail probabilities of a normalized bispectrum of a series under the assumption the model is a linear process with iid innovations.

**Usage**

```
test.linear(series, color = TRUE, detrend = FALSE, main = NULL)
```

**Arguments**

series	the time series (univariate only)
color	if FALSE, the graphic is produced in gray scale
detrend	if TRUE, the series is detrended first
main	if NULL (default), a very nice title is chosen for the plot

**Value**

prob	matrix of tail probabilities - returned invisibly
------	---

**Note**

The null hypothesis is that the data are from a linear process with i.i.d. innovations. Under the null hypothesis, the bispectrum is constant over all frequencies. Chi-squared test statistics are formed in blocks to measure departures from the null hypothesis and the corresponding p-values are displayed in a graphic and returned invisibly. Details are in Hinich, M. and Wolinsky, M. (2005). Normalizing bispectra. *Journal of Statistical Planning and Inference*, 130, 405–411.

**Author(s)**

D.S. Stoffer

## References

You can find demonstrations of astsa capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

## Examples

```
test.linear(nyse) # :(
test.linear(soi) # :)
```

---

timex

*Convert eXtensible Time Series Dates to Decimal Dates*

---

## Description

Takes an 'xts' data file, extracts the dates and converts them to decimal dates without having to have the 'xts' package loaded.

## Usage

```
timex(xts.object)
```

## Arguments

xts.object      an 'xts' data object

## Details

For a data file made using the 'xts' package, this will produce a vector of decimal dates corresponding to the date of each observation even if 'xts' is not loaded. The script is a converter from 'unix time stamp' to decimal dates. For example, the unix time stamp of 2016-04-20 at 00:00:00 UTC is 1461110400. This will be converted to 2016.301 because 2016 was a leap year and April 20 is day 110 of the year using zero-based indexing (Jan 1 is day 0). Thus its decimal date is 2016 + (110/366).

The input object must be an 'xts' object. Note that if dog is an 'xts' data file with columns of time series, `dog[,2]` is NOT unless 'xts' is loaded. Thus, `t <- timex(dog)` will work but `t <- timex(dog[,2])` may not.

## Value

A vector of decimal dates is returned invisibly.

**Note**

We recommend the installation of 'xts' if you are working with time series, but we wanted to have an "option out" in the off chance that it can't (or won't) be installed.

**Author(s)**

D.S. Stoffer

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**Examples**

```
tsplot(timex(lap.xts), lap.xts[, 'Cmort'], col=4, ylab=NA, gg=TRUE)
title('Daily Cardiovascular Mortality')
```

```
DJIA = ts(djia[, 'Close'])
x = cbind(DJIA, Return = diff(log(DJIA)))
tsplot(timex(djia), x, col=5, main="What, Me Worry?")
```

---

trend

*Estimate Trend*

---

**Description**

Estimates the trend (polynomial or lowess) of a time series and returns a graphic of the series with the trend and error bounds (returned invisibly) superimposed.

**Usage**

```
trend(series, order = 1, lowess = FALSE, lowspan = .75, robust = TRUE,
      col = c(4, 6), ylab = NULL, ci = TRUE, results = FALSE, ...)
```

**Arguments**

series	The time series to be analyzed (univariate only).
order	Order of the polynomial used to estimate the trend with a linear default (order=1) unless lowess is TRUE.
lowess	If TRUE, loess from the stats package is used to fit the trend. The default is FALSE.

lowspan	The smoother span used for lowess.
robust	If TRUE (default), the lowess fit is robust.
col	Vector of two colors for the graphic, first the color of the data (default is blue [4]) and second the color of the trend (default is magenta [6]). Both the data and trend line will be the same color if only one value is given.
ylab	Label for the vertical axis (default is the name of the series).
ci	If TRUE (default), pointwise 95% confidence intervals are drawn.
results	For polynomial regression, if TRUE, will print a summary (using <code>ttable</code> ) of the regression results.
...	Other graphical parameters.

### Details

Produces a graphic of the time series with the trend and a .95 pointwise confidence interval superimposed. The trend estimate and the error bounds are returned invisibly.

### Value

Produces a graphic and returns the trend estimate `fit` and error bounds `lwr` and `upr` invisibly (see details) and with the same time series attributes as the input series.

### Author(s)

D.S. Stoffer

### References

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

### See Also

[detrrend](#)

### Examples

```
par(mfrow=2:1)
trend(soi, results=TRUE)
trend(soi, lowess=TRUE, gg=TRUE)
```

---

tspairs	<i>Scatterplot Matrix for Time Series</i>
---------	---

---

**Description**

Produces a matrix of scatterplots with the time series (or a histogram) plotted on the diagonal.

**Usage**

```
tspairs(x, main = NA, pt.col = astsa.col(4, 0.6), pt.size = 1.1, lab.size = 1.25,
        title.size = 1.5, scale = 1, corr = TRUE, smooth = TRUE, lwl = 1, lwc = 2,
        gg = FALSE, hist.diag = TRUE, col.diag = 4, location='topright', ...)
```

**Arguments**

x	multiple time series; use <code>ts.intersect</code> to include lagged values.
main	title (default is no title).
pt.col	point color.
pt.size	point size.
lab.size	label size.
title.size	title size.
scale	multiplier for the overall character expansion ( <code>cex</code> ).
corr	if TRUE (default), the correlations are shown in the scatterplots.
smooth	if TRUE (default), a lowess fit is displayed in the scatterplots.
lwl	width of the lowess line.
lwc	color of the lowess line.
gg	if TRUE, will produce a gris-gris plot (gray graphic interior with white grid lines); the default is FALSE. The grammar of <code>astsa</code> is <code>voodoo</code> .
hist.diag	if TRUE (default), will plot histograms on the diagonal; if FALSE, time plots of the series are displayed instead.
col.diag	color for the diagonal plots.
location	the location of the ACF legend with options "bottomright", "bottom", "bottomleft", "left", "topleft", "top", "topright" (the default), "right" and "center".
...	additional graphic parameters.

**Value**

Returns a matrix of scatterplots with time plots or histograms on the diagonal.

**Note**

Use `lag1.plot` and `lag2.plot` for lag plots. But if some lagged variables are included, use `ts.intersect`. If there are no lagged variables, `cbind` will work to combine individual series.

**Author(s)**

D.S. Stoffer

**References**

You can find demonstrations of astsa capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**See Also**

[lag1.plot](#), [lag2.plot](#)

**Examples**

```
tspairs(diff(log(econ5))[,1:3], col.diag=6, hist=FALSE, pt.size=1.5, lwl=2, gg=TRUE)
```

```
tspairs(ts.intersect(MEI, MEI2), location='top')
```

---

tspplot

*Time Series Plot*


---

**Description**

Produces a nice plot of univariate or multiple time series in one easy line.

**Usage**

```
tspplot(x, y=NULL, main=NULL, ylab=NULL, xlab=NULL, title=NULL, type=NULL, margins=.25,
        omargins=0, ncolm=1, byrow=TRUE, nx=NULL, ny=nx, minor=TRUE, nxm=2, nym=1,
        xm.grid=TRUE, ym.grid=TRUE, col=1, gg=FALSE, spaghetti=FALSE, pch=NULL, lty=1,
        lwd=1, mgpp=0, topper=NULL, addLegend=FALSE, location='topright', boxit=TRUE,
        horiz=FALSE, legend=NULL, llwd=NULL, scale=1, reset.par = TRUE, ...)
```

**Arguments**

x, y	time series to be plotted; if both present, x will be the time index.
main	add a main title - the default is no title.
ylab	y-axis label - the default is the name of the ts object.
xlab	x-axis label - the default is 'Time'.
title	add an individual plot title for multiple plots - the default is no title.
type	type of plot - the default is line.

margins	inches to add (or subtract) to the margins. Input one value to apply to all margins or a vector of length 4 to add (or subtract) to the (bottom, left, top, right) margins.
omargins	inches to add to the outer margins. Input one value to apply to all margins or a vector of length 4 to add to the (bottom, left, top, right) outer margins.
ncolm	for multiple time series, the number of columns to plot.
byrow	for multiple time series - if TRUE (default), plot series row wise; if FALSE, plot series column wise.
nx, ny	number of major cells of the grid in x and y direction. When NULL, as per default, the grid aligns with the tick marks on the corresponding default axis (i.e., tickmarks as computed by axTicks). When NA, no grid lines are drawn in the corresponding direction.
minor, nxm, nym	if minor=TRUE, the number of minor tick marks on x-axis, y-axis. minor=FALSE removes both or set either to 0 or 1 to remove. The default is one minor tick on the x-axis and none on the y-axis.
xm.grid, ym.grid	if TRUE (default), adds grid lines at minor x-axis, y-axis ticks.
col	line color(s), can be a vector for multiple time series.
gg	if TRUE, will produce a gris-gris plot (gray graphic interior with white grid lines); the default is FALSE. The grammar of astsa is voodoo; see <a href="https://www.youtube.com/watch?v=b4J8VrprGE">https://www.youtube.com/watch?v=b4J8VrprGE</a>
spaghetti	if TRUE, will produce a spaghetti plot (all series on same plot).
pch	plot symbols (default is 1, circle); can be a vector for multiple plots.
lty	line type (default is 1, solid line); can be a vector for multiple plots.
lwd	line width (default is 1); can be a vector for multiple plots.
mgpp	this is used to adjust (add to) the mgp graphics parameters settings (?par), which are c(1.6, .6, 0) here; the R default is c(3, 1, 0). This will be helpful in moving an axis label farther from the axis if necessary.
topper	non-negative value to add to the top outer margin; if NULL (default) a suitable value is chosen
addLegend	if TRUE and spaghetti=TRUE, will add a simple legend. If more details are needed, leave this set to FALSE (the default) and use <a href="#">legend</a> directly.
location	if addLegend=TRUE, the location of the legend with options "bottomright", "bottom", "bottomleft", "left", "topleft", "top", "topright" (the default), "right" and "center".
boxit	if TRUE (default), the legend is in a box; if FALSE, no box is drawn.
horiz	if addLegend=TRUE, should the legend be horizontal (default is FALSE - vertical).
legend	if NULL (default), the legend uses names of each time series; otherwise, use to customize legend.
llwd	line width for the legend if different from the plotted lines.
scale	for multiple series, the scale for character expansion.
reset.par	logical; for multiple series, should the graphics parameters be reset after plotting.
...	other graphical parameters; see <a href="#">par</a> .

**Value**

Produces a graphic and returns it invisibly so it can be saved in an R variable with the ability to replay it; see [recordPlot](#).

**Note**

A legend can be added using `addLegend=TRUE` for spaghetti plots only. Spaghetti plots work if `spaghetti=TRUE` and there is more than one series being plotted.

For multiple series, the default for `reset.par` is `TRUE` so that the graphic parameters are reset at the end. For example, if there are 5 plots and `ncol = 2`, the layout will be 3 rows, 2 columns, with an empty spot where a 6th plot would be. If you want to add something in the empty space, set this to `FALSE`, otherwise the graphic parameters are reset and the graphic is finished.

The scale setting is to make sure that axis labels do not get too small for large multifigure plots. While small text can be decent on a screen, it can be too small for publication. The default setting is `scale=1` with larger/smaller values giving larger/smaller text.

**Author(s)**

D.S. Stoffer

**Source**

This is simply base-R plotting of time series with some bells and whistles, and written so that a nice plot can be used in demonstrations (in class or otherwise) with minimal keystrokes.

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**See Also**

[lag1.plot](#), [lag2.plot](#), [tspairs](#),  
[timex](#) for plotting xts data files when xts is not loaded

**Examples**

```
## Not run:  
  
# minimal  
tsplo(soi)  
  
# gris-gris spaghetti  
tsplo(cbind(MEI1, MEI2), spaghetti=TRUE, col=2*2:1, addLegend=TRUE, nym=2, gg=TRUE)
```

```
# gris-gris multiple plot
tsplot(diff(log(econ5)), ncolm=2, gg=TRUE, col=2:6, lwd=2, main='Growth Rates')

# plotting is row-wise unless byrow=FALSE is included
tsplot(climhyd, ncolm=2, gg=TRUE, col=2:7, lwd=2, xlab=c(rep(NA,4), rep('Time',2)),
       scale=.85, ylab=NA, title=colnames(climhyd), las=1)

# missing data - try this with ggplot
tsplot(blood, type='o', col=c(4,6,3), pch=19, scale=.95, gg=TRUE,
       title = colnames(blood), xlab = c(NA, NA, 'day'),
       ylab = c('log(cells/\u03BCL)', 'log(\u03BCL)', '\u0025 red blood cells' ))

## End(Not run)
```

---

ttable	<i>t-table summary for an lm object</i>
--------	---

---

## Description

Works like `summary` for an `lm` object but adds AIC, AICc, and BIC to the output with the option to display VIFs.

## Usage

```
ttable(obj, digits = 4, vif = FALSE, ...)
```

## Arguments

obj	an object of class "lm" as a result of a call to <code>lm</code> .
digits	the (approximate) number of significant digits to use when printing.
vif	if TRUE, variance inflation factors are printed if applicable; default is FALSE.
...	further arguments passed to or from other methods.

## Details

Produces a t-table for an `lm` object much like `print.summary.lm` with added information including AIC, AICc, BIC, and VIF (if requested and if applicable), to the output. The output is rounded and there are no significance stars. In fact, there are no stars at all.

TO REPEAT THE WARNING ON USING 'lm' FOR TIME SERIES: Considerable care must be taken:

- Include `na.action = NULL` in the `lm` call to avoid stripping the time series attributes from the variables before the regression is done.
- If any lagged or differenced variables are used in 'lm', the series must be lined up first. In this case, prepare a data frame using `ts.intersect(..., dframe = TRUE)`; e.g.,

```
mydata = ts.intersect(M = cmort, P4 = lag(part,-4), dframe=TRUE)
fit     = lm(M ~ P4, na.action=NULL, data=mydata)
ttable(fit)
```

### Value

Prints a typical t-table with additional information as mentioned in the details. The p-values are two-sided. Also silently returns the same values as described in [summary.lm](#).

### Author(s)

D.S. Stoffer

### Source

Built using `print.summary.lm` from the 'stats' package.

### References

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

### See Also

[summary.lm](#), [print.summary.lm](#), [lm](#)

### Examples

```
fit1 = lm(cmort~ time(cmort) + tempr + I(tempr^2))
ttable(fit1, vif=TRUE)
# if you center `tempr`, the squared term doesn't change
temp = tempr - mean(tempr)
fit2 = lm(cmort~ time(cmort) + temp + I(temp^2))
ttable(fit2, vif=TRUE)
```

---

unemp

*U.S. Unemployment*

---

### Description

Monthly U.S. Unemployment series (1948-1978, n = 372)

### Usage

```
data(unemp)
```

### Format

The format is: Time-Series [1:372] from 1948 to 1979: 235 281 265 241 201 ...

### References

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

### See Also

[UnempRate](#)

---

UnempRate

*U.S. Unemployment Rate*

---

### Description

Monthly U.S. unemployment rate in percent unemployed (Jan, 1948 - Nov, 2016, n = 827)

### Format

The format is: Time-Series [1:827] from 1948 to 2017: 4 4.7 4.5 4 3.4 3.9 3.9 3.6 3.4 2.9 ...

### Source

<https://data.bls.gov/timeseries/LNU0400000/>

**References**

You can find demonstrations of astsa capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**See Also**

[unemp](#)

---

USpop

*U.S. Population - 1900 to 2010*

---

**Description**

U.S. Population by official census, every ten years from 1900 to 2010.

**Format**

The format is: Time-Series [1:12] from 1900 to 2010: 76 92 106 123 132 ...

**Details**

The census from 2020 is not included in this data set because, by many accounts, it was a nightmare (<https://www.npr.org/2022/01/15/1073338121/2020-census-interference-trump>) due to the COVID-19 pandemic coupled with the fact that the Census Bureau is in the Department of Commerce, and its head is appointed by and reports directly to the POTUS, who at the time was DJ tRump: "Historians rank Trump among worst presidents in US history ..." (<https://www.businessinsider.com/historia>

The data with the 2020 census is in [USpop20](#). Note that the two data files differ a bit, presumably because they are revised ad infinitum. Both data sets were obtained from the Census Bureau's website.

**Source**

<https://www.census.gov/>

**References**

You can find demonstrations of astsa capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**See Also**[USpop20](#)

---

[USpop20](#)*U.S. Population - 1900 to 2020*

---

**Description**

U.S. Population by official census, every ten years from 1900 to 2020, in millions.

**Format**

The format is: Time-Series [1:13] from 1900 to 2020: 76.2 92.2 106 122.8 132.2 ...

**Details**

The census from 2020 is included in this data set, but the results from 2020 are questionable. By many accounts, it was a nightmare (<https://www.npr.org/2022/01/15/1073338121/2020-census-interference-trump>) due to the COVID-19 pandemic coupled with the fact that the Census Bureau is in the Department of Commerce, and its head is appointed by and reports directly to the POTUS, who at the time was DJ tRump: "Historians rank Trump among worst presidents in US history ... " (<https://www.businessinsider.com/historians-rank-trump-among-worst-presidents-us-history-c-span-2020>)

The data without the 2020 census is in [USpop](#). Note that the two data files differ a bit, presumably because they are revised ad infinitum. Both data sets were obtained from the Census Bureau's website.

**Source**

<https://www.census.gov/>

**References**

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

**See Also**[USpop](#)

## Examples

```
# the regression
t = time(USpop20) - 1960
reg = lm( USpop20~ poly(t, 10, raw=TRUE) )

# the prediction curve
b = as.vector(coef(reg))
t = 1900:2044
X = outer(t - 1960, 0:10, FUN = "^")
pred = X %*% b

# the plot
tsplot(t, pred, ylab="Population", xlab='Year', cex.main=1, col=4,
       main="U.S. Population by Official Census")
points(time(USpop20), USpop20, pch=21, bg=rainbow(13), cex=1.25)
mtext(bquote('\u00D7~10^6'), side=2, line=1.5, adj=1, cex=.8)
```

---

varve

*Annual Varve Series*

---

## Description

Sedimentary deposits from one location in Massachusetts for 634 years, beginning nearly 12,000 years ago.

## Format

The format is: Time-Series [1:634] from 1 to 634: 26.3 27.4 42.3 58.3 20.6 ...

## Source

Antevs, E. (1928). The Last Glaciation with Special Reference to the Ice Retreat in Northeastern North America. *Amer. Geog. Soc. Res. Series*, No. 17, 292p.

Shumway and Verosub (1992). State space modeling of paleoclimatic time series. *Proceedings of the 5th International Meeting on Statistical Climatology*, June, 1992, University of Toronto, 139p.

## References

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

---

WBC

*White Blood Cell Levels*

---

### Description

WBC: Measurements made for 91 days on the three variables, log(white blood count) [WBC], log(platelet) [PLT] and hematocrit [HCT]. Missing data code is 0 (zero).

### Format

The format is: Time-Series [1:91] from 1 to 91: 2.33 1.89 2.08 1.82 1.82 ...

### Details

See Examples 6.1 and 6.9 for more details.

### Source

Jones, R.H. (1984). Fitting multivariate models to unequally spaced data. In *Time Series Analysis of Irregularly Observed Data*, pp. 158-188. E. Parzen, ed. Lecture Notes in Statistics, 25, New York: Springer-Verlag.

### References

You can find demonstrations of `astsa` capabilities at [FUN WITH ASTSA](#).

The most recent version of the package can be found at <https://github.com/nickpoison/astsa/>.

In addition, the News and ChangeLog files are at <https://github.com/nickpoison/astsa/blob/master/NEWS.md>.

The webpages for the texts and some help on using R for time series analysis can be found at <https://nickpoison.github.io/>.

### See Also

[blood](#), [HCT](#), [PLT](#)

# Index

## \* **attribute**

Months, [83](#)

## \* **color**

astsa.col, [19](#)

## \* **datasets**

ar1miss, [13](#)

arf, [14](#)

BCJ, [26](#)

beamd, [27](#)

birth, [28](#)

blood, [28](#)

bnrf1ebv, [29](#)

bnrf1hvs, [30](#)

cardox, [30](#)

chicken, [33](#)

climhyd, [33](#)

cmort, [34](#)

cpg, [35](#)

djia, [36](#)

EBV, [39](#)

econ5, [40](#)

ENSO, [43](#)

EQ5, [44](#)

eqexp, [45](#)

EXP6, [49](#)

flu, [52](#)

fmri, [53](#)

fmri1, [54](#)

gas, [55](#)

gdp, [56](#)

GDP23, [56](#)

gnp, [57](#)

GNP23, [58](#)

gtemp.month, [60](#)

gtemp\_both, [61](#)

gtemp\_land, [61](#)

gtemp\_ocean, [62](#)

Hare, [63](#)

HCT, [64](#)

hor, [64](#)

jj, [65](#)

lap, [75](#)

lap.xts, [76](#)

lead, [78](#)

Lynx, [78](#)

MEI, [80](#)

MEI2, [82](#)

nyse, [87](#)

oil, [87](#)

part, [88](#)

PLT, [89](#)

polio, [90](#)

prodn, [93](#)

qinfl, [93](#)

qintr, [94](#)

rec, [96](#)

sales, [97](#)

salmon, [97](#)

salt, [98](#)

saltemp, [99](#)

sleep1, [109](#)

sleep2, [110](#)

so2, [111](#)

soi, [112](#)

soiltemp, [112](#)

sp500.gr, [113](#)

sp500w, [114](#)

speech, [118](#)

star, [120](#)

sunspotz, [122](#)

tempr, [126](#)

unemp, [137](#)

UnempRate, [137](#)

USpop, [138](#)

USpop20, [139](#)

varve, [140](#)

WBC, [141](#)

## \* **distribution**

- QQnorm, 95
- \* **hplot**
  - QQnorm, 95
- \* **manip**
  - dna2vector, 37
- \* **math**
  - matrixpwr, 79
  - polyMul, 90
- \* **package**
  - astsa-package, 4
- \* **print**
  - ttable, 135
- \* **ts**
  - acf1, 6
  - acf2, 7
  - acfm, 8
  - ar.boot, 10
  - ar.mcmc, 11
  - arma.check, 14
  - arma.spec, 16
  - ARMAtoAR, 18
  - astsa-package, 4
  - autoParm, 21
  - autoSpec, 23
  - bart, 25
  - ccf2, 31
  - detrend, 35
  - EM, 41
  - ESS, 47
  - ffbs, 50
  - Kfilter, 66
  - Ksmooth, 68
  - lag1.plot, 71
  - lag2.plot, 72
  - LagReg, 74
  - mvspec, 84
  - pre.white, 91
  - sarima, 99
  - sarima.for, 102
  - sarima.sim, 104
  - SigExtract, 107
  - spec.ic, 114
  - specenv, 116
  - ssm, 118
  - stoch.reg, 120
  - SV.mcmc, 123
  - SV.mle, 125
  - test.linear, 127
  - timex, 128
  - trend, 129
  - tspairs, 131
  - tsplot, 132
  - %^(matrixpwr), 79
  - acf1, 6, 8, 9, 32
  - acf2, 7, 7, 9, 32
  - acfm, 7, 8, 8, 32
  - ar, 115
  - ar.boot, 10
  - ar.mcmc, 11
  - ar1miss, 13
  - arf, 14
  - arma.check, 14, 17
  - arma.spec, 16, 16
  - ARMAtoAR, 16, 18
  - ARMAtoMA, 16
  - astsa (astsa-package), 4
  - astsa-package, 4
  - astsa.col, 19
  - autoParm, 21, 24
  - autoSpec, 22, 23
  - bart, 25
  - BCJ, 26
  - beamd, 27
  - birth, 28
  - blood, 28, 64, 89, 141
  - bnrf1ebv, 29
  - bnrf1hvs, 30
  - cardox, 30
  - ccf2, 7–9, 31, 92
  - chicken, 33
  - climhyd, 33
  - cmort, 34
  - cpg, 35
  - detrend, 35, 130
  - djia, 36
  - dna2vector, 37, 40, 116, 117
  - EBV, 39
  - econ5, 40
  - EM, 41
  - ENSO, 43, 44, 81, 82, 112
  - EQ5, 44
  - EQcount, 45

- eqexp, [44](#), [45](#), [49](#)
- ESS, [47](#)
- EXP6, [49](#)
- FDR, [49](#)
- ffbs, [50](#)
- flu, [52](#)
- fmri, [53](#), [54](#)
- fmri1, [54](#)
- gas, [55](#), [88](#)
- GDP, [56–58](#)
- GDP (GDP23), [56](#)
- gdp, [56](#), [57](#), [58](#)
- GDP23, [56](#)
- GNP, [56](#), [57](#)
- GNP (GNP23), [58](#)
- gnp, [56](#), [57](#), [57](#), [58](#)
- GNP23, [58](#)
- Grid, [58](#)
- grid, [59](#)
- gtemp.month, [60](#)
- gtemp\_both, [61](#), [62](#)
- gtemp\_land, [61](#), [61](#), [62](#)
- gtemp\_ocean, [61](#), [62](#), [62](#)
- Hare, [63](#), [79](#)
- HCT, [29](#), [64](#), [89](#), [141](#)
- hor, [64](#)
- hsv, [19](#)
- jj, [65](#)
- Kfilter, [42](#), [51](#), [66](#), [70](#)
- Ksmooth, [41](#), [42](#), [68](#), [68](#)
- lag1.plot, [71](#), [73](#), [131](#), [132](#), [134](#)
- lag2.plot, [72](#), [72](#), [131](#), [132](#), [134](#)
- LagReg, [74](#)
- lap, [34](#), [75](#), [77](#), [88](#), [111](#), [127](#)
- lap.xts, [34](#), [76](#), [76](#), [88](#), [127](#)
- lead, [78](#), [97](#)
- legend, [133](#)
- lm, [135](#), [136](#)
- Lynx, [63](#), [78](#)
- lynx, [79](#)
- matrixpwr, [79](#)
- MEI, [44](#), [80](#), [82](#), [112](#)
- MEI2, [44](#), [81](#), [82](#), [112](#)
- Months, [83](#)
- mvspec, [84](#)
- nyse, [87](#)
- oil, [55](#), [87](#)
- par, [133](#)
- part, [88](#)
- pie, [19](#)
- PLT, [29](#), [64](#), [89](#), [141](#)
- polio, [90](#)
- polyMul, [90](#)
- pre.white, [32](#), [91](#)
- print.summary.lm, [135](#), [136](#)
- prodn, [93](#)
- qinfl, [93](#), [94](#)
- qintr, [94](#), [94](#)
- QQnorm, [95](#)
- rec, [96](#), [112](#)
- recordPlot, [134](#)
- sales, [78](#), [97](#)
- salmon, [97](#)
- salt, [98](#), [99](#)
- saltemp, [98](#), [99](#)
- sarima, [99](#), [102](#), [103](#)
- sarima.for, [101](#), [102](#)
- sarima.sim, [16](#), [101](#), [104](#)
- scatter.hist, [106](#)
- SigExtract, [107](#)
- sleep1, [109](#), [111](#)
- sleep2, [110](#), [110](#)
- so2, [111](#)
- soi, [43](#), [44](#), [81](#), [82](#), [97](#), [112](#)
- soiltemp, [112](#)
- sp500.gr, [113](#)
- sp500w, [114](#)
- spec.ar, [115](#)
- spec.ic, [48](#), [114](#)
- specenv, [38](#), [116](#)
- speech, [118](#)
- ssm, [118](#)
- star, [120](#)
- stoch.reg, [120](#)
- summary.lm, [136](#)
- sunspots, [122](#)
- sunspotz, [122](#)

SV.mcmc, [123](#), [126](#)

SV.mle, [124](#), [125](#)

tempr, [126](#)

test.linear, [127](#)

timex, [128](#), [134](#)

trend, [36](#), [129](#)

tspairs, [72](#), [73](#), [131](#), [134](#)

tsplot, [6](#), [7](#), [9](#), [32](#), [95](#), [132](#)

ttable, [130](#), [135](#)

unemp, [137](#), [138](#)

UnempRate, [137](#), [137](#)

USpop, [138](#), [139](#)

USpop20, [138](#), [139](#), [139](#)

varve, [140](#)

WBC, [29](#), [64](#), [89](#), [141](#)