

R FUNCTIONS FOR TIME SERIES ANALYSIS

Here are some helpful R functions for time series analysis. They belong from *stats*, *tseries*, *ast* and *lmtest* packages and grouped by their goal.

INPUT

cycle(): gives the positions in the cycle of each observation (stats)
deltat(): returns the time interval between observations (stats)
end(): extracts and encodes the times the last observation were taken (stats)
frequency(): returns the number of samples per unit time (stats)
read.ts(): reads a time series file (tseries)
start(): extracts and encodes the times the first observation were taken (stats)
time(): creates the vector of times at which a time series was sampled (stats)
ts(): creates time-series objects (stats)
window(): is a generic function which extracts the subset of the object 'x' observed between the times 'start' and 'end'. If a frequency is specified, the series is then re-sampled at the new frequency (stats)

TS DECOMPOSITION

decompose(): decomposes a time series into seasonal, trend and irregular components using moving averages. Deals with additive or multiplicative seasonal component (stats)
filter(): linear filtering on a time series (stats)
HoltWinters(): computes Holt-Winters Filtering of a given time series (stats)
sfilter(): removes seasonal fluctuation using a simple moving average (ast)
spectrum(): estimates the spectral density of a time series (stats)
stl(): decomposes a time series into seasonal, trend and irregular components using 'loess' (stats)
tsr(): decomposes a time series into trend, seasonal and irregular. Deals with additive and multiplicative components (ast)

TESTS

adf.test(): computes the Augmented Dickey-Fuller test for the null that 'x' has a unit root (tseries)
Box.test(): computes the Box-Pierce or Ljung-Box test statistic for examining the null hypothesis of independence in a given time series (stats)
bds.test(): computes and prints the BDS test statistic for the null that 'x' is a series of i.i.d. random variables (tseries)
bptest(): performs the Breusch-Pagan test for heteroskedasticity of residuals (lmtest)
dwtest(): performs the Durbin-Watson test for autocorrelation of residuals (lmtest)
jarque.bera.test(): Jarque-Bera test for normality (tseries)
kpss.test(): computes KPSS test for stationarity (tseries)
shapiro.test(): Shapiro-Wilk Normality Test (stats)

STOCHASTIC MODELS

ar(): fits an autoregressive time series model to the data, by default selecting the complexity by AIC (stats)
arima(): fits an ARIMA model to a univariate time series (stats)
arima.sim(): simulate from an ARIMA model (stats)
arma(): fits an ARMA model to a univariate time series by conditional least squares (tseries)
garch(): fits a Generalized Autoregressive Conditional Heteroscedastic GARCH(p, q) time series model to the data by computing the maximum-likelihood estimates of the conditionally normal model (tseries)

GRAPHICS

lag.plot: plots time series against lagged versions of themselves. Helps visualizing "auto-dependence" even when auto-correlations vanish (stats)
monthplot(): plots a seasonal (or other) subseries of a time series (stats)
plot.ts(): plotting time-series objects (stats)
seaplot(): plotting seasonal sub-series or profile (ast)

seqplot.ts(): plots a two time series on the same plot frame (tseries)

tsdiag(): a generic function to plot time-series diagnostics (stats)

ts.plot(): plots several time series on a common plot. Unlike 'plot.ts' the series can have a different time bases, but they should have the same frequency (stats)

MISCELLANEOUS

acf(), *pacf()*, *ccf()*: the function 'acf' computes (and by default plots) estimates of the autocovariance or autocorrelation function. Function 'pacf' is the function used for the partial autocorrelations. Function 'ccf' computes the cross-correlation or cross-covariance of two univariate series (stats)

diff.ts(): returns suitably lagged and iterated differences (stats)

lag(): computes a lagged version of a time series, shifting the time base back by a given number of observations (stats)