str {utils}

Compactly display the internal **str**ucture of an **R** object, a diagnostic function and an alternative to [summary](http://127.0.0.1:21197/library/utils/help/summary) (and to some extent, [dput](http://127.0.0.1:21197/library/utils/help/dput)). Ideally, only one line for each ‘basic’ structure is displayed. It is especially well suited to compactly display the (abbreviated) contents of (possibly nested) lists. The idea is to give reasonable output for **any** **R** object. It calls [args](http://127.0.0.1:21197/library/utils/help/args) for (non-primitive) function objects.

strOptions() is a convenience function for setting [options](http://127.0.0.1:21197/library/utils/help/options)(str = .), see the examples.

<http://www.stat.pitt.edu/stoffer/tsa3/index.html>

<http://www.stat.pitt.edu/stoffer/tsa3/R_toot.htm>

If you're a Matlab (or similar) user, you may think jj is an 84 × 1 vector, but it's not. It has order and length, but no dimensions (no rows, no columns). R calls these kinds of objects "vectors" so you have to be careful. In R, "matrices" have dimensions but "vectors" do not.

jj[1] # *the first element*

[1] 0.71

jj[84] # *the last element*

[1] 11.61

jj[1:4] # *the first 4 elements*

[1] 0.71 0.63 0.85 0.44

jj[-(1:80)] # *everything EXCEPT the first 80 elements*

[1] 16.20 14.67 16.02 11.61

length(jj) # *the number of elements*

[1] 84

dim(jj) # *but no dimensions ...*

NULL

nrow(jj) # *... no rows*

NULL

ncol(jj) # *... and no columns*

NULL

#-- *if you want it to be a column vector (in R, a matrix), an easy way to go is:*

jjm = as.matrix(jj)

dim(jjm)

[1] 84 1

◊ On ARMAX: If you want to fit an ARMAX model you have to do it via a state space model... the details are in Chapter 6 of the text. Note that you can't use xreg in arima() to fit an ARMAX model (it was never claimed you could). You can only use it to do regression with autocorrelated errors.