The Other Packages Gallery

David J. Scott

February 2, 2016

Contents

1 Introduction

This document represents a test of the functions in **xtable** which deal with other packages.

The first step is to load the package and set some options for this document.

```
library(xtable)
options(xtable.floating = FALSE)
options(xtable.timestamp = "")
options(width = 60)
set.seed(1234)
```

2 The packages spdep, splm, and sphet

Code for supporting these packages and most of the examples used in this section was originally provided by Martin Gubri (`martin.gubri@framasoft.org`).

2.1 The package spdep

First load the package and create some objects.

```
library(spdep)
```

```
## Loading required package:  sp
## Loading required package:  Matrix
```

```
data("oldcol", package = "spdep")
COL.lag.eig <- lagsarlm(CRIME ~ INC + HOVAL, data = COL.OLD[],
                        nb2listw(COL.nb))
class(COL.lag.eig)
```

```
## [1] "sarlm"
```

```
COL.errW.GM <- GMerrorsar(CRIME ~ INC + HOVAL, data = COL.OLD,
                          nb2listw(COL.nb, style = "W"),
                          returnHcov = TRUE)
class(COL.errW.GM)
```

```
## [1] "gmsar"
```

```
COL.lag.stsls <- stsls(CRIME ~ INC + HOVAL, data = COL.OLD,
                       nb2listw(COL.nb))
class(COL.lag.stsls)
```

```
## [1] "stsls"
```

```
p1 <- predict(COL.lag.eig, newdata = COL.OLD[45:49,],
              listw = nb2listw(COL.nb))
```

```
## Warning in predict.sarlm(COL.lag.eig, newdata = COL.OLD[45:49, ], listw = nb2listw(COL.nb)):
some region.id are both in data and newdata
```

```
class(p1)
```

```
## [1] "sarlm.pred"
```

```
p2 <- predict(COL.lag.eig, newdata = COL.OLD[45:49,],
              pred.type = "trend", type = "trend")
```

```
## Warning in predict.sarlm(COL.lag.eig, newdata = COL.OLD[45:49, ], pred.type =
"trend", :  some region.id are both in data and newdata
```

```
#type option for retrocompatibility with spdep 0.5-92
class(p2)
```

```
## [1] "sarlm.pred"
```

```
imp.exact <- impacts(COL.lag.eig, listw = nb2listw(COL.nb))
class(imp.exact)
```

```
## [1] "lagImpact"
```

```
imp.sim <- impacts(COL.lag.eig, listw = nb2listw(COL.nb), R = 200)
class(imp.sim)
```

```
## [1] "lagImpact"
```

sarlm objects

There is an `xtable` method for objects of this type.

```
xtable(COL.lag.eig)
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	45.0793	7.1773	6.28	0.0000
INC	-1.0316	0.3051	-3.38	0.0007
HOVAL	-0.2659	0.0885	-3.00	0.0027

The method for `xtable` actually uses the summary of the object, and an identical result is obtained when using the summary of the object, even if the summary contains more additional information.

```
xtable(summary(COL.lag.eig, correlation = TRUE))
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	45.0793	7.1773	6.28	0.0000
INC	-1.0316	0.3051	-3.38	0.0007
HOVAL	-0.2659	0.0885	-3.00	0.0027

This same pattern applies to the other objects from this group of packages.

Note that additional prettying of the resulting table is possible, as for any table produced using `xtable`. For example using the **booktabs** package we get:

```
print(xtable(COL.lag.eig), booktabs = TRUE)
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	45.0793	7.1773	6.28	0.0000
INC	-1.0316	0.3051	-3.38	0.0007
HOVAL	-0.2659	0.0885	-3.00	0.0027

gmsar objects

```
xtable(COL.errW.GM)
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	62.5138	5.1213	12.21	0.0000
INC	-1.1283	0.3397	-3.32	0.0009
HOVAL	-0.2970	0.0957	-3.10	0.0019

stsls objects

```
xtable(COL.lag.stsls)
```

	Estimate	Std. Error	t value	Pr(>|t|)
Rho	0.4546	0.1851	2.46	0.0141
(Intercept)	43.7934	10.9522	4.00	0.0001
INC	-1.0007	0.3839	-2.61	0.0091
HOVAL	-0.2655	0.0919	-2.89	0.0038

sarlm.pred objects

`xtable` has a method for predictions of `sarlm` models.

```
xtable(p1)
```

	fit	trend	signal
1045	30.24	19.15	11.09
1046	25.72	16.09	9.63
1047	24.65	15.49	9.16
1048	16.80	5.94	10.86
1049	17.69	6.60	11.09

This method transforms the `sarlm.pred` objects into data frames, allowing any number of attributes vectors which may vary according to predictor types.

```
xtable(p2)
```

	fit
1045	19.15
1046	16.09
1047	15.49
1048	5.94
1049	6.60

lagImpact objects

The `xtable` method returns the values of direct, indirect and total impacts for all the variables in the model. The class `lagImpact` has two different sets of attributes according to if simulations are used. But the `xtable` method always returns the three components of the non-simulation case.

```
xtable(imp.exact)
```

	Direct	Indirect	Total
INC	-1.09	-0.73	-1.81
HOVAL	-0.28	-0.19	-0.47

```
xtable(imp.sim)
```

	Direct	Indirect	Total
INC	-1.09	-0.73	-1.81
HOVAL	-0.28	-0.19	-0.47

spautolm objects

The need for an `xtable` method for `spautolm` was expressed by Guido Schulz (`schulzgu@student.hu-berlin.de`), who also provided an example of an object of this type. The required

code was implemented by David Scott (`d.scott@auckland.ac.nz`).

First create an object of the required type.

```
library(spdep)
example(NY_data)
```

```
## Field name:  'Z600701190' changed to:  'Z600701190.1'
## Field name:  'Z600701190' changed to:  'Z600701190.2'
## Field name:  'Z600701200' changed to:  'Z600701200.1'
## Field name:  'Z600701210' changed to:  'Z600701210.1'
## Field name:  'Z600701210' changed to:  'Z600701210.2'
## Field name:  'Z600701220' changed to:  'Z600701220.1'
## Field name:  'Z600701270' changed to:  'Z600701270.1'
## Field name:  'Z600701320' changed to:  'Z600701320.1'
## Field name:  'Z600701330' changed to:  'Z600701330.1'
## Field name:  'Z605303010' changed to:  'Z605303010.1'
## Field name:  'Z605303010' changed to:  'Z605303010.2'
## Field name:  'Z605303040' changed to:  'Z605303040.1'
## Field name:  'Z605303040' changed to:  'Z605303040.2'
## Field name:  'Z605303050' changed to:  'Z605303050.1'
## Field name:  'Z606700170' changed to:  'Z606700170.1'
## Field name:  'Z606700360' changed to:  'Z606700360.1'
## Field name:  'Z606700560' changed to:  'Z606700560.1'
## Field name:  'Z606700610' changed to:  'Z606700610.1'
## Field name:  'Z606700610' changed to:  'Z606700610.2'
## Field name:  'Z606701100' changed to:  'Z606701100.1'
## Field name:  'Z606701110' changed to:  'Z606701110.1'
## Field name:  'Z606701120' changed to:  'Z606701120.1'
## Field name:  'Z606701120' changed to:  'Z606701120.2'
## Field name:  'Z606701120' changed to:  'Z606701120.3'
## Field name:  'Z606701140' changed to:  'Z606701140.1'
## Field name:  'Z606701520' changed to:  'Z606701520.1'
## Field name:  'Z606701650' changed to:  'Z606701650.1'
## Field name:  'Z606701680' changed to:  'Z606701680.1'
## Field name:  'Z606701690' changed to:  'Z606701690.1'
```

```
spautolmOBJECT <- spautolm(Z ~ PEXPOSURE + PCTAGE65P,data = nydata,
                           listw = listw_NY, family = "SAR",
                           method = "eigen", verbose = TRUE)
summary(spautolmOBJECT, Nagelkerke = TRUE)
```

```
class(spautolmOBJECT)
```

```
## [1] "spautolm"
```

```
xtable(spautolmOBJECT,
       display = c("s",rep("f", 3), "e"), digits = 4)
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-0.9128	0.1258	-7.2541	4.0457e-13
PEXPOSURE	0.0915	0.0453	2.0204	4.3346e-02
PCTAGE65P	3.8294	0.6313	6.0655	1.3151e-09

2.2 The package splm

First load the package and create some objects.

```
library(splm)
```

```
## Note:  the specification for S3 class "family" in package 'MatrixModels' seems
equivalent to one from package 'lme4':  not turning on duplicate class definitions
for this class.
```

```
data("Produc", package = "plm")
data("usaww",  package = "splm")
fm <- log(gsp) ~ log(pcap) + log(pc) + log(emp) + unemp
respatlag <- spml(fm, data = Produc, listw = mat2listw(usaww),
                  model="random", spatial.error="none", lag=TRUE)
class(respatlag)
```

```
## [1] "splm"
```

```
GM <- spgm(log(gsp) ~ log(pcap) + log(pc) + log(emp) + unemp, data = Produc,
           listw = usaww, moments = "fullweights", spatial.error = TRUE)
```

```
## Note:  method with signature 'diagonalMatrix#Matrix' chosen for function 'kronecker',
##  target signature 'ddiMatrix#dgCMatrix'.
##  "ANY#sparseMatrix" would also be valid
## Note:  method with signature 'dsparseMatrix#dsparseMatrix' chosen for function
'kronecker',
##  target signature 'dtTMatrix#dgCMatrix'.
##  "TsparseMatrix#sparseMatrix" would also be valid
```

```
class(GM)
```

```
## [1] "splm"
```

```
imp.spml <- impacts(respatlag, listw = mat2listw(usaww, style = "W"), time = 17)
class(imp.spml)
```

```
## [1] "lagImpact"
```

splm objects

```
xtable(respatlag)
```

	Estimate	Std. Error	t-value	Pr(>|t|)
(Intercept)	1.6581	0.1507	11.00	0.0000
log(pcap)	0.0129	0.0249	0.52	0.6037
log(pc)	0.2256	0.0216	10.43	0.0000
log(emp)	0.6708	0.0264	25.39	0.0000
unemp	-0.0058	0.0009	-6.50	0.0000

```
xtable(GM)
```

	Estimate	Std. Error	t-value	Pr(>|t|)
log(pcap)	0.0043	0.0253	0.17	0.8652
log(pc)	0.2145	0.0233	9.22	0.0000
log(emp)	0.7831	0.0280	27.99	0.0000
unemp	-0.0026	0.0011	-2.43	0.0152

The `xtable` method works the same on impacts of `splm` models.

```
xtable(imp.spml)
```

	Direct	Indirect	Total
log(pcap)	0.01	0.00	0.02
log(pc)	0.23	0.04	0.27
log(emp)	0.68	0.12	0.80
unemp	-0.01	-0.00	-0.01

2.3 The package sphet

First load the package and create some objects.

```
library(sphet)
```

```
##
## Attaching package:  'sphet'
##
## The following object is masked from 'package:splm':
##
##     listw2dgCMatrix
```

```
data("columbus", package = "spdep")
listw <- nb2listw(col.gal.nb)
data("coldis", package = "sphet")
res.stsls <- stslshac(CRIME ~ HOVAL + INC, data = columbus, listw = listw,
                      distance = coldis, type = 'Triangular')
class(res.stsls)
```

```
## [1] "sphet" "stsls"
```

```
res.gstsls <- gstslshet(CRIME ~ HOVAL + INC, data = columbus, listw = listw)
class(res.gstsls)
```

```
## [1] "sphet"  "gstsls"
```

```
imp.gstsls <- impacts(res.gstsls, listw = listw)
class(imp.gstsls)
```

```
## [1] "lagImpact"
```

sphet objects

```
xtable(res.stsls)
```

	Estimate	SHAC St.Er.	t-value	Pr(>|t|)
Wy	0.4546	0.1651	2.75	0.0059
(Intercept)	44.1164	8.0757	5.46	0.0000
HOVAL	-0.2695	0.1787	-1.51	0.1315
INC	-1.0077	0.5119	-1.97	0.0490

```
xtable(res.gstsls)
```

	Estimate	Std. Error	t-value	Pr(>|t|)
(Intercept)	44.1241	7.5067	5.88	0.0000
HOVAL	-0.2756	0.1769	-1.56	0.1193
INC	-0.9875	0.4602	-2.15	0.0319
lambda	0.4529	0.1433	3.16	0.0016
rho	0.0648	0.3051	0.21	0.8317

`sphet` also provides a method for computing impacts.

```
xtable(imp.gstsls)
```

	Direct	Indirect	Total
HOVAL	-0.29	-0.21	-0.50
INC	-1.05	-0.76	-1.80

3 The zoo package

```
library(zoo)
```

```
##
## Attaching package:  'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
xDate <- as.Date("2003-02-01") + c(1, 3, 7, 9, 14) - 1
as.ts(xDate)
```

Time Series: Start = 1 End = 5 Frequency = 1 [1] 12084 12086 12090 12092 12097

```
x <- zoo(rnorm(5), xDate)
xtable(x)
```

	Value
12084	-0.81
12085	
12086	2.55
12087	
12088	
12089	
12090	-0.13
12091	
12092	0.79
12093	
12094	
12095	
12096	
12097	-1.74

```
tempTs <- ts(cumsum(1 + round(rnorm(100), 0)),
             start = c(1954, 7), frequency = 12)
tempTable <- xtable(tempTs, digits = 0)
tempTable
```

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1954							2	3	3	3	5	6
1955	6	9	12	13	14	15	16	16	18	19	18	18
1956	20	22	22	22	23	25	27	28	29	29	31	31
1957	30	32	32	33	34	35	36	36	36	36	38	40
1958	42	41	43	44	45	46	47	47	48	50	51	52
1959	56	57	58	59	60	61	61	62	64	66	67	67
1960	67	68	69	70	70	72	74	75	76	76	77	78
1961	79	80	81	83	85	86	87	90	91	92	91	92
1962	93	94	94	94	95	96	96	98	99	99		

```
tempZoo <- as.zoo(tempTs)
xtable(tempZoo, digits = 0)
```

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1954							2	3	3	3	5	6
1955	6	9	12	13	14	15	16	16	18	19	18	18
1956	20	22	22	22	23	25	27	28	29	29	31	31
1957	30	32	32	33	34	35	36	36	36	36	38	40
1958	42	41	43	44	45	46	47	47	48	50	51	52
1959	56	57	58	59	60	61	61	62	64	66	67	67
1960	67	68	69	70	70	72	74	75	76	76	77	78
1961	79	80	81	83	85	86	87	90	91	92	91	92
1962	93	94	94	94	95	96	96	98	99	99		

4 The survival package

```
library(survival)
test1 <- list(time=c(4,3,1,1,2,2,3),
              status=c(1,1,1,0,1,1,0),
              x=c(0,2,1,1,1,0,0),
              sex=c(0,0,0,0,1,1,1))
coxFit <- coxph(Surv(time, status) ~ x + strata(sex), test1)
xtable(coxFit)
```

	coef	exp(coef)	se(coef)	z	p
x	0.80	2.23	0.82	0.98	0.33