Package ‘erer’

June 26, 2012

Title Empirical Research in Economics with R

Version 1.2

Date 2010-11-15 (first built); 2012-6-26 (last)

Author Changyou Sun <csun@cfr.msstate.edu>

Maintainer Changyou Sun <csun@cfr.msstate.edu>

Depends R (>= 2.14.0), systemfit, lmtest, tseries, ggplot2

Description This package contains functions and datasets for the book of 'Empirical Research in Economics: Growing up with R' by Dr. Changyou Sun. These functions can calculate marginal effects for a binary probit or logit model, estimate static and dynamic Almost Ideal Demand System (AIDS) models, and conduct event analysis.

License GPL (>= 2)

LazyLoad yes

R topics documented:

`erer-package` *Empirical Research in Economics with R*

Description

This package contains functions and datasets for the book of 'Empirical Research in Economics: Growing up with R' by Dr. Changyou Sun. These functions can calculate marginal effects for a binary probit or logit model, estimate static and dynamic Almost Ideal Demand System (AIDS) models, and conduct event analysis.

Details

Package:	erer
Type:	Package
Version:	1.2
Date:	2010-11-15 (first built); 2012-6-26 (last)
Depends:	R (>= 2.14.0), systemfit, lmtest, tseries, ggplot2
License:	GPL (>= 2)
LazyLoad:	yes

Author(s)

Changyou Sun <csun@cfr.msstate.edu>

`aiData` *Transforming Raw Data for Static AIDS Model*

Description

This function transforms import values and quantities into a data format that are needed for a static AIDS model.

Usage

```
aiData(x, label, label.tot = "WD", prefix.value = "v",
  prefix.quant = "q", start = NULL, end = NULL, ...)
```

Arguments

x	raw time series data such as daBedRaw.
label	names of supplying countries; this can be as long as needed.
label.tot	names of the world total (default label is "WD").
prefix.value	prefix for value variables.
prefix.quant	prefix for quantity varibles.
start	start date for the transformed time series; this can be used to select a smaller window; the default is the start date of the raw data x.
end	end date for the transformed time series.
...	additional arguments to be passed.

Details

This transforms raw import data into a format needed for a static AIDS model. This separation of data prepraration from model fitting allows greater flexibility in using aiStaFit in estimating a static AIDS model.

Value

Return a list object with two components:

out	a time series object ready for static AIDS models.
share	a data frame object of the share data.
price	a data frame object of the price data.
m	a vector of the total expenditure.
call	a record of the system call; this allows update.default to be used.

Author(s)

Changyou Sun (<csun@cfr.msstate.edu>)

References

Wan, Y., C. Sun, and D.L. Grebner. 2010. Analysis of import demand for wooden beds in the United States. Journal of Agricultural and Applied Economics 42(4):643-658.

See Also

aiStaFit; daBedRaw; daBed.

Examples

```
data(daBedRaw)
imp8 <- aiData(x = daBedRaw,
  label = c("CN", "VN", "ID", "MY", "CA", "BR", "IT"),
  label.tot = "WD", prefix.value = "v", prefix.quant = "q",
  start = c(2001, 1), end = c(2008, 12), frequency = 12)
imp4 <- update(imp8, label = c("CN", "VN", "ID"))
imp5 <- update(imp4, label = c("CN", "VN", "ID", "MY"))
imp8; imp4; imp5
```

```
dat8 <- imp8$out

dum  <- ts(0, start = start(dat8), end = end(dat8), frequency = 12)
dum1 <- replace(dum, time(dum) == 2003+(10-1)/12, 1)
dum2 <- replace(dum, time(dum) == 2004+(7 -1)/12, 1)
dum3 <- replace(dum, time(dum) == 2005+(1 -1)/12, 1)
daTest <- ts.union(dat8, dum1, dum2, dum3)
colnames(daTest) <- c(colnames(dat8), "dum1", "dum2", "dum3")

data(daBed)
identical(daBed, daTest)
```

---

aiDiag *Diagnostic Statitics for Static or Dynamic AIDS Model*

Description

Report a set of diagnostic statistics for static or dynamic AIDS models

Usage

```
aiDiag(x, digits = 3, ...)
```

Arguments

x	an object of class aiFit from the function of aiStaFit or aiDynFit.
digits	number of digits used in rounding outputs.
...	additional arguments to be passed.

Details

Compute several diagnostic statistics for each equation in a AIDS model. Tests includes are BG, BP, RESET, and JB. See the reference paper for detail.

Value

Return a data frame object with the statistics and p values for the four tests by equation.

Author(s)

Changyou Sun (<csun@cfr.msstate.edu>)

References

Wan, Y., C. Sun, and D.L. Grebner. 2010. Analysis of import demand for wooden beds in the United States. Journal of Agricultural and Applied Economics 42(4):643-658.

See Also

aiStaFit; aiDynFit.

Examples

```
# see the examples for 'aiDynFit'.
```

`aiDynFit` *Fitting a Dynamic AIDS Model*

Description

Estimate a dynamic AIDS model for a system.

Usage

```
aiDynFit(w, dum.dif = FALSE, ...)
```

Arguments

`w`	a object of class `aiStaFit`.
`dum.dif`	a logical value (default of FALSE) of whether to take a difference on the dummy variables passed from `w`.
`...`	additional arguments to be passed.

Details

This estimates a dynamic AIDS model. The residuals from the statis AIDS model are included. As it is programmed now, only one lag is allowed for the share variables on the right-hand side.

Value

Return a list object of class "aiFit" and "aiDynFit" with the following components:

`w`	a object of class `aiStaFit`.
`y`	data for fitting the static AIDS model, passed down by `w`.
`dum.dif`	a logical value (default of FALSE) of whether to take a difference on the dummy variables passed from `w`.
`daDyn`	data for fitting the dynamic AIDS model.
`share`	names of shares by commodity, used as depedent variables.
`price`	names of prices by commodity, used as independent variables.
`expen`	names of expenditure variable.
`shift`	names of the shifters.
`omit`	names of the omitted share variable.
`nOmit`	position of the omitted share variable in the name of share variable.
`hom`	a logical value of homogeneity test.
`sym`	a logical value of symmetry test.
`nShare`	number of share variables.
`nExoge`	number of exogenous variables (lagged share, residual, expenditure, and shifters).
`nParam`	number of parameters in one equation.
`nTotal`	number of parameters in the whole system estimated.
`formula`	formula for estimating the system.
`res.matrix`	restriction matrix for `hom` or `sym`, or both.
`res.rhs`	right-hand values for tests of `hom` or `sym`, or both.
`est`	the dynamic AIDS model estimated.

Author(s)

Changyou Sun (<csun@cfr.msstate.edu>)

References

Wan, Y., C. Sun, and D.L. Grebner. 2010. Analysis of import demand for wooden beds in the United States. Journal of Agricultural and Applied Economics 42(4):643-658.

See Also

aiStaFit; aiDiag; aiElas; summary.aiFit.

Examples

```
# --- Step 1: Read data
data(daExp, daBedRaw, daBed)

# --- Step 2: Hausman Test
# 2.1 Getting started with a static AIDS model
sh  <- c("sCN", "sVN", "sID", "sMY", "sCA", "sBR", "sIT", "sRW")
pr  <- c("lnpCN", "lnpVN", "lnpID", "lnpMY",
         "lnpCA", "lnpBR", "lnpIT", "lnpRW")
du3 <- c("dum1","dum2","dum3")
rSta <- aiStaFit(y = daBed, share = sh, price = pr, shift = du3,
  expen = "rte", omit = "sRW", hom = TRUE, sym = TRUE)
summary(rSta)

# 2.2 The final Hausman test and new data
(dg <- daExp[, "dg"])
rHau <- aiStaHau(x = rSta, instr = dg, choice = FALSE)
names(rHau); colnames(rHau$daHau); colnames(rHau$daFit); rHau
two.exp <- rHau$daFit[, c("rte", "rte.fit")]
bsStat(two.exp, digits = 4)
plot(data.frame(two.exp)); abline(a = 0, b = 1)
daBedFit <- rHau$daFit

# --- Step 3: Static and dynamic AIDS models
# 3.1 Diagnostics and coefficients
hSta  <- update(rSta, y = daBedFit, expen = "rte.fit")
hSta2 <- update(hSta, hom = FALSE, sym = FALSE)
hSta3 <- update(hSta, hom = FALSE, sym = TRUE)
hSta4 <- update(hSta, hom = TRUE,  sym = FALSE)
lrtest(hSta2$est, hSta$est)
lrtest(hSta2$est, hSta3$est)
lrtest(hSta2$est, hSta4$est)

hDyn  <- aiDynFit(hSta)
hDyn2 <- aiDynFit(hSta2); lrtest(hDyn2$est, hDyn$est)
hDyn3 <- aiDynFit(hSta3); lrtest(hDyn2$est, hDyn3$est)
hDyn4 <- aiDynFit(hSta4); lrtest(hDyn2$est, hDyn4$est)

(table.2 <- rbind(aiDiag(hSta), aiDiag(hDyn)))
(table.3 <- summary(hSta))
(table.4 <- summary(hDyn))
```

```
# 3.2 Elasticity calculation
es <- aiElas(hSta); esm <- es$marsh
ed <- aiElas(hDyn); edm <- ed$marsh
esm2 <- data.frame(c(esm[1:2, 2], esm[3:4, 3],
  esm[5:6, 4], esm[7:8, 5], esm[9:10, 6], esm[11:12, 7],
  esm[13:14, 8], esm[15:16, 9]))
edm2 <- data.frame(c(edm[1:2, 2], edm[3:4, 3],
  edm[5:6, 4], edm[7:8, 5], edm[9:10, 6], edm[11:12, 7],
  edm[13:14, 8], edm[15:16, 9]))
eEM <- cbind(es$expen, esm2, ed$expen[2], edm2)
colnames(eEM) <- c("Country", "LR.expen", "LR.Marsh",
  "SR.expen", "SR.Marsh")
(table.5 <- eEM[-c(15:16),])
(table.6a <- es$hicks[-c(15:16), -9])
(table.6b <- ed$hicks[-c(15:16), -9])
```

`aiElas` *Computing Elasticity for Static or Dynamic AIDS Models*

Description

Calculate expenditure elasticity, Marshalllian price elasticity, Hicksian price elasticity, and their variances for static or dynamic AIDS Models.

Usage

```
aiElas(z, digits = 3, ...)
```

Arguments

`z`	an object of class `aiFit` from the function of `aiStaFit` or `aiDynFit`.
`digits`	number of digits used in rounding outputs.
`...`	additional arguments to be passed.

Details

Calculate expenditure elasticity, Marshalllian price elasticity, and Hicksian price elasticity for static or dynamic AIDS Models. The related variance, t-ratio, p-value, and significance are also reported.

Value

Return a list object with the following components:

`name`	name of the share variables; the omitted share name is the last one.
`expen`	expenditure elasticity and related statistics.
`marsh`	Marshalllian price elasticity and related statistics.
`hicks`	Hicksian price elasticity and related statistics.

Author(s)

Changyou Sun (<`csun@cfr.msstate.edu`>)

References

Wan, Y., C. Sun, and D.L. Grebner. 2010. Analysis of import demand for wooden beds in the United States. Journal of Agricultural and Applied Economics 42(4):643-658.

See Also

aiStaFit; aiDynFit.

Examples

```
# see the examples for 'aiDynFit'.
```

---

aiStaFit — *Fitting a Static AIDS Model*

Description

Estimate a static AIDS model for a system.

Usage

```
aiStaFit(y, share, price, expen, shift = NULL, omit = NULL,
  hom = TRUE, sym = TRUE, ...)
```

Arguments

y	a multiple time series data.
share	names of the share variables.
price	names of the price variables.
expen	name of the expenditure variables.
shift	names of the shifter variables.
omit	name of the share variable omitted; if not supplied, this is the last one of share.
hom	a logical value of homogeneity test.
sym	a logical value of symmetry test.
...	additional arguments to be passed.

Details

This estimates a static AIDS model. The data supplied should be in the final format.

Value

Return a list object of class "aiFit" and "aiStaFit" with the following components:

y	data for fitting the static AIDS model.
share	names of the share variables.
price	names of the price variables.
expen	name of the expenditure variables.

shift	names of the shifter variables.
omit	name of the share variable omitted; if not supplied, this is the last one of share.
nOmit	position of the omitted share variable in the name of share variable.
hom	a logical value of homogeneity test.
sym	a logical value of symmetry test.
nShare	number of share variables.
nExoge	number of exogenous variables (lagged share, residual, expenditure, and shifters).
nParam	number of parameters in one equation.
nTotal	number of parameters in the whole system estimated.
formula	formula for estimating the system.
res.matrix	restriction matrix for hom or sym, or both.
res.rhs	right-hand values for tests of hom or sym, or both.
est	the static AIDS model estimated.
call	a record of the system call; this allows update.default to be used.

Author(s)

Changyou Sun (<csun@cfr.msstate.edu>)

References

Wan, Y., C. Sun, and D.L. Grebner. 2010. Analysis of import demand for wooden beds in the United States. Journal of Agricultural and Applied Economics 42(4):643-658.

See Also

aiDiag; aiElas; summary.aiFit; aiDynFit.

Examples

```
# see the examples for 'aiDynFit'.
```

---

aiStaHau — *Conducting a Hausman Test on a Static AIDS Model*

Description

Conduct a Hausman test on a static AIDS model and report the result of likelihood ratio test.

Usage

```
aiStaHau(x, instr, choice = FALSE, ...)
```

Arguments

`x`	an object of class `aiStaFit` from a static AIDS model.
`instr`	a single time series data as instrument for the expenditure variable in AIDS model.
`choice`	a logical value of whether to take a difference on the right-hand `price` and `instr` variables.
`...`	additional arguments to be passed.

Details

Conduct a Hausman test on a static AIDS model and report the result of likelihood ratio test. Note that logarithm is taken on every variable in the auxiliary regression. These variables are the real total expenditure and its lagged value, instrumental variable, and the price variables.

Value

Return a data frame object with the statistics and p values for the four tests by equation.

`daHau`	data used in estimating the Hausman test.
`formuHau`	formula for estimating the Hausman test.
`regHau`	regression for the Hausman test.
`daFit`	revised data with the fitted value of expenditure included.
`aiBase`	the base static AIDs model estimated.
`aiHaus`	the reestimated static AIDS model using the fitted value of expenditure.
`ratio`	result of the likelihood ration test for the Hausman test.

Author(s)

Changyou Sun (<csun@cfr.msstate.edu>)

References

Wan, Y., C. Sun, and D.L. Grebner. 2010. Analysis of import demand for wooden beds in the United States. Journal of Agricultural and Applied Economics 42(4):643-658.

See Also

`aiStaFit`; `print.aiFit`.

Examples

```
# see the examples for 'aiDynFit'.
```

bsFormu *Generating Formula for Models*

Description

Generate a single formula for models like `lm` or a list of formula for models like `systemfit`.

Usage

```
bsFormu(name.y, name.x, intercept = TRUE, ...)
```

Arguments

`name.y`	a character vector of variables names for dependent variables; when the length is more than one, there will a list of formula generated for each variable in the name.
`name.x`	a character vector of indepedent variables.
`intercept`	a logical value (default of TRUE) of whether to include intercept or not.
`...`	additional arguments to be passed.

Details

This function can generate a single formula for simple model like `lm` or a list of formula for systems (`systemfit`. Note that the right-hand side variables are the same for each dependent variable. If different, a `for` loop can be added by users to address that, as demonstrated by the example below.

Value

a single formula object or a list of formula objects.

Author(s)

Changyou Sun (<csun@cfr.msstate.edu>)

Examples

```
# fake data
y  <- c("y")
ym <- c("y1", "y2", "y3")
x <- c("x")
xAll <- c("x", "xx", "xxx", "xxxx")

bsFormu(name.y = y, name.x = x)
bsFormu(name.y = ym, name.x = xAll)
fm.ym <- bsFormu(name.y = ym, name.x = xAll, intercept = FALSE)
fm.ym

# If independent variables differ by equation,
# add a loop to address the differentiation.
xInd <- c("x1", "x2", "x3")
fm.ym <- list()
```

```
for (i in 1:length(ym)) {
    ny <- ym[i]
    nx <- c(xInd[i], xAll)
    fm.ym[[i]] <- bsFormu(name.y = ny, name.x = nx, intercept = FALSE)
}
fm.ym

# real data
data(daIns)
(xx <- colnames(daIns)[-c(1, 14)])
fm.ins <- bsFormu(name.y = "Y", name.x = xx, intercept = TRUE)
fm.ins
(ra <- glm(formula = fm.ins,
          family = binomial(link="logit"),
          data = daIns, x = TRUE))
```

---

bsLag *Lagged Time Series*

Description

Generate a set of lagged time series for time series data.

Usage

```
bsLag(h, lag, prefix = "", var.name, suffix = ".t_",
    include.orig = TRUE, by.lag = FALSE, ...)
```

Arguments

`h`	time series data
`lag`	number of lags
`prefix`	prefix for the name of lagged time series.
`var.name`	varible name of the lagged time series.
`suffix`	suffix of the name of lagged time series.
`include.orig`	logical value (default of TRUE) of whether to include the original series (i.e., lag zero) in the final output.
`by.lag`	logical value (default of FALSE) of whether to order the column by variable (FALSE) or by lag (TRUE).
`...`	additional arguments to be passed.

Details

The input data can be a single time series or a set of multiple time series data. The output is a set of lagged time series with the specified lag dimension. All the series are aligned with the shortest window so the loss of observations is equal to `lag`. The original series (e.g., without lag but just loss of beginning observations) can be included or excluded by setting the logical value of `include.org`.

The name of the output data is composed of four parts: `prefix`, `var.name`, `suffix`, and an index number of `lag`. Users can control the first three parts only because the lag number is added automatically. `prefix` and `suffix` can be fixed for all the output series. `var.name` provides some flexibility when `bsLag` is used within a function and the variable name is unknown *a priori*.

The column of the output can be ordered either by the variable name (e.g., diff.GA.t_0, diff.GA.t_1, diff.ND.t_0, diff.ND.t_1), or by the lag order ((e.g., diff.GA.t_0, diff.ND.t_0, diff.GA.t_1, diff.ND.t_1).

Value

Return a multiple time series object.

Author(s)

Changyou Sun (<csun@cfr.msstate.edu>)

Examples

```
# simple example
h1 <- ts(data=cbind(1:24), start=c(2001, 1), frequency=12)
h2 <- ts(data=cbind(1:24, 25:48), start=c(2001, 1), frequency=12)
h3 <- ts(data=cbind(1:4, 5:8, 9:12), start=c(2001, 1), frequency=4)
colnames(h2) <- c("aa", "bb")
colnames(h3) <- c("cc", "dd", "ee")
h1; h2; h3

bsLag(h=h1, lag=0, prefix="", suffix=".t_")
bsLag(h=h1, lag=2, prefix="price.", var.name="fl", suffix=".t_")
bsLag(h=h1, lag=2, prefix="price.", var.name="fl", suffix=".t_", by.lag=TRUE)
bsLag(h=h1, lag=23, prefix="price.", suffix=".t_", include.orig=FALSE)

bsLag(h=h2, lag=4, prefix="", suffix=".t_", include.orig = TRUE)
bsLag(h=h2, lag=4, prefix="", suffix=".t_", include.orig = FALSE)
bsLag(h=h2, lag=4, prefix="", suffix=".t_", include.orig = FALSE, by.lag=TRUE)
bsLag(h=h2, lag=0, prefix="", var.name=c("nc", "sc"), suffix=".t_")

bsLag(h=h3, lag=2, prefix="", suffix=".t_", include.orig=FALSE)
bsLag(h=h3, lag=1, prefix="", var.name=c("nd", "sd", "mi"), suffix=".lag.")
bsLag(h=h3, lag=2, prefix="NY.", suffix=".t_", by.lag=TRUE)
bsLag(h=h3, lag=3, prefix="NY.", suffix=".t_", include.orig=FALSE)

# with real data
data(daBedRaw)
small <- daBedRaw[, c("vCN", "qCN")]
(lag.small <- bsLag(h=small, lag=4))
colnames(lag.small)

resid <- residuals(lm(qCN ~ vCN, data = small))
res <- ts(resid, start=start(small), end=end(small),
    frequency=tsp(small)[3])
lag.res <- bsLag(h=res, lag=2, prefix="resid.", var.name="china")
lag.res
```

---

bsStat *Summary of Basic Statistics*

Description

Calcluate basic statistics of data.

Usage

```
bsStat(y, two = NULL, digits = c(2, 1), ...)
```

Arguments

y	input data for summary statistics.
two	a logical value of whether to report the correlation and summary statistics separately; if NULL and the number of variables is less than 11, its value will be set to TRUE.
digits	digits for the output data, one for correlation coefficients and the other for mean and others; if a single scalar is supplied, it will be used for both.
...	additional arguments to be passed.

Details

Two set of summary statistics are generated. One is correlation coefficients and the other is mean, minimum, maximum, standard error, and number of observations. When two is unspecified and the number of variables is bigger than ten, the two sets are reported separately; otherwise, it is reported as a single data frame object.

Value

A dataframe or list of the summary statistics.

Author(s)

Changyou Sun (<csun@cfr.msstate.edu>)

Examples

```
data(daIns)
(sum.daIns <- bsStat(y=daIns, digits=c(3,2)))
```

---

bsTab *Generating Pretty Statistical Tables*

Description

Format statistics from regressions into pretty outputs

Usage

```
bsTab(w,
  need = c("1T", "1E", "2T", "2E", "3T", "3E", "4T", "4E", "5"),
  wrap.TE = c("(", "", "["),
  add.sig = c("coef", "TE"),
  percent = c(0.01, 0.05, 0.10),
  symbol = c("***", "**", "*", ""),
  digits = c(3, 3, 3, 2), ... )
```

Arguments

`w` statistical results from regression models; an object of class `glm`, `lm`, and `systemfit` can be supplied directly, or a data frame with at least four columns with the sequence of estimates, errors, t-values, and p-values.

`need` a choice of output formats; default of `1T` is one column with t ratio and significance symbols; 1 to 5 is the number of columns; T is t ratios; E is standard errors. This argument must be a character string.

`wrap.TE` parentheses, none, or brackets can be used to enclose t ratios or standard errors; default value is parenthses for one-column format and none for other formats.

`add.sig` a chacter string to indicate where to add the significance symbol, either to the coefficents (`"coef"`) or the t-value and error (`"TE"`).

`percent` percentage values used to categorize p values.

`symbol` symbols used to represent p-value categories; the default values can be changed to symbols like a, b, c, or different combinations of *.

`digits` digits for outputs; the default values are 3, 3, 3, and 2 for estimate, error, t value, and p value, correspondingly. A single value like 4 can be supplied and it will be recycled for all of them.

`...` additional arguments to be passed.

Details

Format statistics from regressions into tables that are often reported in economic journals. The column of 'Variable' in the outuput is the row names of the input data so the raw data should contain meaningful rownames. Besides the variable name column, the maximum number of output is five columns: estimate, error, t ratio, p value, and significance. `wrap.TE` and `add.sig` are only valid for column widths of 1 and 2.

Value

A dataframe of statistical results.

Author(s)

Changyou Sun (<csun@cfr.msstate.edu>)

Examples

```
# a simulated data
tes <- data.frame(est = c(4, 56, 12), err = c(0.3, 0.56, 0.789),
  t.rat = c(2.56, 7.9, 1.2), p.val = c(0.002, 0.23, 0.061))
tes
bsTab(tes)
bsTab(w = tes, need = "2E")

# real data
data(daIns)
ra <- glm(formula = Y ~ Injury + HuntYrs + Nonres +
                    Lspman + Lnong + Gender + Age +
                    Race + Marital + Edu + Inc + TownPop,
          family = binomial(link="logit"),
          data = daIns, x = TRUE)
```

```
(ca <- data.frame(summary(ra)$coefficients))

# an object of class 'glm' as input
bsTab(w = ra, add.sig = "TE")
bsTab(w = ra, wrap.TE = "[")
bsTab(w = ra, need = "5")
bsTab(w = ra, need = "4T", wrap.TE = "[")
final <- bsTab(w = ra, need = "3T",
  percent = c(0.01, 0.05, 0.10),
  symbol = c("a", "b", "c", ""), digits = 4)
final
print(final, right = FALSE)

# any matrix with at least four columns can be supplied
cbind(bsTab(ca), bsTab(ra))
```

`daBed` *Transformed Wooden Beds Import Data for Static AIDS Models*

Description

This data set contains transformed values related to wooden beds imports by the United States from January 2001 to December 2008. There are 96 observations and 20 variables.

`sCN`	monthly import share of wooden beds from China
`sVN`	monthly import share of wooden beds from Vietnam
`sID`	monthly import share of wooden beds from Indonesia
`sMY`	monthly import share of wooden beds from Malaysia
`sCA`	monthly import share of wooden beds from Canada
`sBR`	monthly import share of wooden beds from Brazil
`sIT`	monthly import share of wooden beds from Italy
`sRW`	monthly import share of wooden beds from the rest of world
`rte`	real total expenditure in logarithm
`lnpCN`	monthly import price of wooden beds from China in logarithm
`inpVN`	monthly import price of wooden beds from Vietnam in logarithm
`lnpID`	monthly import price of wooden beds from Indonesia in logarithm
`lnpMY`	monthly import price of wooden beds from Malaysia in logarithm
`lnpCA`	monthly import price of wooden beds from Canada in logarithm
`inpBR`	monthly import price of wooden beds from Brazil in logarithm
`lnpIT`	monthly import price of wooden beds from Italy in logarithm
`lnpRW`	monthly import price of wooden beds from the rest of world in logarithm
`dum1`	a pulse dummy variable (1 for October 2003, 0 otherwise)
`dum2`	a pulse dummy variable (1 for July 2004, 0 otherwise)
`dum3`	a pulse dummy variable (1 for January 2005, 0 otherwise)

Usage

```
data(daBed)
```

Format

Monthly time series from Janurary 2001 to December 2008 with 96 observations for each of the 20 variables.

Details

This is the transformated data set for static AIDS model. The transformation detail is described in Wan et al. (2010).

Source

U.S. ITC, 2010. Interactive tariff and trade data web. http://dataweb.usitc.gov (Assecced on March 1, 2010).

References

Wan, Y., C. Sun, and D.L. Grebner. 2010. Analysis of import demand for wooden beds in the United States. Journal of Agricultural and Applied Economics 42(4):643-658.

See Also

aiStaFit; daBedRaw.

Examples

```
data(daBed)
class(daBed); dim(daBed); colnames(daBed)
daBed
```

---

daBedRaw *Wooden Beds Import Data*

Description

This data set contains a mulitiple time series related to wooden beds imports by the United States. The time covered is January 1996 to December 2008 with 156 observations. There are 34 variables in total: 17 import values (dollars) and 17 import quantities (dollars / piece). In total, 16 countries are covered and the world total is also reported.

vBR	cost-insurance-freight import values in dollar from Brazil
vCA	cost-insurance-freight import values in dollar from Canada
vCN	cost-insurance-freight import values in dollar from China
vDK	cost-insurance-freight import values in dollar from Denmark
vFR	cost-insurance-freight import values in dollar from France
vHK	cost-insurance-freight import values in dollar from Hong Kong
vIA	cost-insurance-freight import values in dollar from India
vID	cost-insurance-freight import values in dollar from Indonesia
vIT	cost-insurance-freight import values in dollar from Italy
vMY	cost-insurance-freight import values in dollar from Malaysia
vMX	cost-insurance-freight import values in dollar from Mexico
vPH	cost-insurance-freight import values in dollar from Philippines
vTW	cost-insurance-freight import values in dollar from Taiwan

vTH	cost-insurance-freight import values in dollar from Thailand
vUK	cost-insurance-freight import values in dollar from United Kingdom
vVN	cost-insurance-freight import values in dollar from Vietnam
vWD	cost-insurance-freight import values in dollar from World in total
qBR	quantity in piece from Brazil
qCA	quantity in piece from Canada
qCN	quantity in piece from China
qDK	quantity in piece from Denmark
qFR	quantity in piece from France
qHK	quantity in piece from Hong Kong
qIA	quantity in piece from India
qID	quantity in piece from Indonesia
qIT	quantity in piece from Italy
qMY	quantity in piece from Malaysia
qMX	quantity in piece from Mexico
qPH	quantity in piece from Philippines
qTW	quantity in piece from Taiwan
qTH	quantity in piece from Thailand
qUK	quantity in piece from United Kingdom
qVN	quantity in piece from Vietnam
qWD	quantity in piece from World in total

Usage

```
data(daBedRaw)
```

Format

Monthly time series from Janurary 1996 to December 2008 with 156 observations for each of the 34 variables.

Details

Under the Harmonized Tariff Schedule (HTS) system, the commodity of wooden beds is classified as HTS 9403.50.9040. The monthly cost-insurance-freight values in dollar and quantities in piece are reported by country from U.S. ITC (2010).

Source

U.S. ITC, 2010. Interactive tariff and trade data web. http://dataweb.usitc.gov (Assecced on March 1, 2010).

References

Wan, Y., C. Sun, and D.L. Grebner. 2010. Analysis of import demand for wooden beds in the United States. Journal of Agricultural and Applied Economics 42(4):643-658.

See Also

aiStaFit; daBed.

Examples

```
data(daBedRaw)
class(daBedRaw); dim(daBedRaw); colnames(daBedRaw)
```

---

daExp *Expenditure Data for a Hausman Test in AIDS Model*

Description

This data set contains seven monthly times series for expenditure from 2001 to 2008.

pinc	Billions of dollars, personal income
dpi	Billions of dollars, disposable personal income
pce	Billions of dollars, personal consumption expenditures
dg	Billions of dollars, Personal consumption expenditures for durable goods
rdpi	Billions of dollars, real disposable personal income
rpce	Billions of dollars, real personal consumption expenditures
rdg	Billions of dollars, real personal consumption expenditures for durable goods

Usage

```
data(daExp)
```

Format

Monthly time series from Janurary 2001 to December 2008 with 96 observations for each of the seven variables.

Details

This is the data set for conducting a Hausman test in a static AIDS model, as detailed in Wan et al. (2010). The test focuses on whether the expenditure variable in a AIDS model is exogenous or not. Each of the seven expenditure data can be used as an instrumental variable in an auxiliary regression.

Source

Federal Reserve Bank of St. Louis. Economic Data - Fred. Internet site: http://stlouisfed.org (Accessed February 25, 2010).

References

Wan, Y., C. Sun, and D.L. Grebner. 2010. Analysis of import demand for wooden beds in the United States. Journal of Agricultural and Applied Economics 42(4):643-658.

Examples

```
data(daExp)
class(daExp); dim(daExp); colnames(daExp)
daExp
```

daIns *Liability Insurance Coverage for Hunters and Anglers in Mississippi*

Description

This data set contains a survey result about liability insurance purchase decision by hunters and anglers in Mississippi. There are 1653 observations for 14 variables.

Y	Binary dependent variable = 1 if had liability insurance; 0 otherwise
Injury	Times of bodily injuries or property damages in the past three years
HuntYrs	Years of hunting
Nonres	Dummy = 1 if nonresidents; 0 if Mississippi residents
Lspman	Dummy = 1 if purchased the license of resident sportsman; 0 otherwise
Lnong	Dummy = 1 if purchased the license of nonresident all game; 0 otherwise
Gender	Dummy = 1 if male; 0 otherwise
Age	Age of the hunter or angler
Race	Dummy = 1 if Caucasian; 0 otherwise
Marital	Dummy = 1 if married; 0 otherwise
Edu	Years of education
Inc	Household income in 2004 (1,000 dollars)
TownPop	Population size of the residence town (1,000)
FishYrs	Years of fishing

Usage

```
data(daIns)
```

Format

A cross sectional data with 1653 observations and 14 variables.

Details

The data set is from a telephone survey conducted in 2005 in Mississippi.

Source

Sun, C., S. Pokharel, W.D. Jones, S.C. Grado, and D.L. Grebner. 2007. Extent of recreational incidents and determinants of liability insurance coverage for hunters and anglers in Mississippi. Southern Journal of Applied Forestry 31(3):151-158.

Examples

```
data(daIns)
class(daIns); dim(daIns)
head(daIns); tail(daIns)

ra <- glm(formula = Y ~ Injury + HuntYrs + Nonres +
                    Lspman + Lnong + Gender + Age +
                    Race + Marital + Edu + Inc + TownPop,
          family = binomial(link="logit"),
```

```
          data = daIns, x = TRUE, y= TRUE)
names(ra); summary(ra)

(ins.me <- maBina(w = ra))
(ins.mt <- maTrend(q=ins.me, nam.c="Age", nam.d="Nonres"))
plot(ins.mt)
```

---

daPe *Program Effectiveness of a New Method of Teaching Economics*

Description

This data set contains the evaluation results of a new program of teaching in economics. There are 32 observations for 4 variables.

grade	a binary variable indicating grade increase (1) and decrease (0) after participation.
gpa	a continous variable measuring studens' grade point average.
tuce	a continous variable measuring students' scores on an economics test.
psi	a binary variable indicating whether a student participates the program or not.

Usage

```
data(daPe)
```

Format

A data frame of cross sectional data with 32 observations and 4 variables.

Details

Evaluation results on 32 students of the impact of a new teaching methods.

Source

Spector, L.C., and M. Mazzeo. 1980. Probit analysis and economic education. Journal of Economic Education 11(2):37-44.

Examples

```
data(daPe)
dim(daPe)
summary(daPe)
daPe
```

evReturn *Estimating Abnormal Return from Event Analysis*

Description

Conduct an event analysis and estimate abnormal returns over time and across firms.

Usage

```
evReturn(y, firm, event.date, y.date = "date",
  index = "sp500", event.win = 3, est.win = 250, digits = 4, ...)
```

Arguments

`y`	a data frame object with one column for date, return series by firms, a return series for a stock market index, and a return series for a risk free asset.
`firm`	a character vector of firm names; this is the name of the return series in `y`.
`event.date`	event dates for each firm as specified in `firm`; this should be a numerical vector and can match the values in `y$y.date`; if event dates are the same for all the firms, this can be specificed as a single number.
`y.date`	a character value for the column name of date in `y`.
`index`	a character value for the column name of index in `y`.
`event.win`	the one-side width of event window in days; the default value of 3 corresponds to a 7-day window (i.e., 3 + 1 + 3).
`est.win`	the width of estimation window in days.
`digits`	number of digits used to format outputs.
`...`	additional arguments to be passed.

Details

This is the core function for event analysis. It estimates a market model by firm and then calculate abnormal returns by firm and over time. The time series of stock returns have irregular time frequency because of varying trading days. Thus, the time dimension is explicitly specified as a `y.date` column in the data of `y`.

Value

Return a list object of class "evReturn" with the following components:

`y`	a data frame of raw return data.
`y.date`	a character value for the column name of date in `y`..
`firm`	a character vector of firm names.
`N`	the number of firms.
`index`	a character value for the column name of index in `y`.
`event.date`	event dates for each firm as specified in `firm`.
`event.win`	the one-side width of event window in days.
`event.width`	total number of days in an event window.

est.win	the width of estimation window in days..
daEst	data used to estimate the market model for the last firm as specified in codefirm.
daEve	data over the event window for the last firm.
ra	fitted market model for the last firm.
digits	number of digits used to format outputs.
reg	regression coefficients by firm.
abr	abnormal returns by day over the event window and by firm.
abc	average abnormal returns across firms.
call	a record of the system call; this allows `update.default` to be used.

Author(s)

Changyou Sun (<csun@cfr.msstate.edu>)

References

Mei, B., and C. Sun. 2008. Event analysis of the impact of mergers and acquisitions on the financial performance of the U.S. forest products industry. Forest Policy and Economics 10(5):286-294.

See Also

`print.evReturn`; `plot.evReturn`; `evRisk`.

Examples

```
# see Mei and Sun (2008).
```

---

evRisk *Risk Evaluation for Event Analysis*

Description

Conduct a risk analysis by firm and evaluate the change of risk before and after an event. The model used is the Captial Asset Pricing Model.

Usage

```
evRisk(x, m = 50, r.free = "tbill", ...)
```

Arguments

x	a object from `evReturn`.
m	the number of days before and after the event date for estimating CAPM.
r.free	the column name of risk free asset in `y`.
...	additional arguments to be passed.

Details

This fits CAPM for each firm and reports the statistics for alpha, beta, and gamma. The statistics of gamma reveal the change of risk before and after the event.

Value

Return a list object of class "evReturn" with the following components:

x	a object from evReturn.
daEst	data used to estimate CAPM for the last firm as specified in codefirm.
rb	fitted CAPM for the last firm.
reg	regression coefficients by firm.

Author(s)

Changyou Sun (<csun@cfr.msstate.edu>)

References

Mei, B., and C. Sun. 2008. Event analysis of the impact of mergers and acquisitions on the financial performance of the U.S. forest products industry. Forest Policy and Economics 10(5):286-294.

See Also

evReturn; print.evRisk.

Examples

```
# see Mei and Sun (2008).
```

---

head   *Return the first or last part of time series data*

Description

Return the first of last parts of an object of time series data.

Usage

```
  ## S3 method for class 'ts'
head(x, n = 5, ...)
  ## S3 method for class 'ts'
tail(x, n = 5, ...)
```

Arguments

x	input time seires data.
n	a single integer for the length or row of returned data
...	additional arguments to be passed.

Details

The data can be an univariate or multivariate time series data.

Value

An object like x but generally smaller.

Author(s)

Changyou Sun (<csun@cfr.msstate.edu>)

Examples

```
h1 <- ts(data=cbind(1:24), start=c(2001, 1), frequency=12)
h2 <- ts(data=cbind(1:24, 25:48), start=c(2001, 1), frequency=12)
h3 <- ts(data=cbind(1:4, 5:8, 9:12), start=c(2001, 1), frequency=4)
colnames(h2) <- c("aa", "bb")
colnames(h3) <- c("cc", "dd", "ee")
h1; h2; h3

h1; head(h1); tail(h1, 28)
h2; head(h2); tail(h2, 50)
h3; head(h3, 2); tail(h3); tail(h3, 8)

data(daBed); head(daBed); tail(daBed)
```

---

maBina *Marginal Effect for Binary Probit and Logit Model*

Description

This function calculates marginal effects for a binary probit or logit model and their standard errors.

Usage

```
maBina(w, x.mean = TRUE, rev.dum = TRUE, digits = 3, ...)
```

Arguments

`w`	a binary probit or logit model object estimated from `glm()`.
`x.mean`	a logical value (default of TRUE) of whether to calculate marginal effects at the means of independent variables. If FALSE, marginal effects are calculated for each observation and then averaged.
`rev.dum`	a logical value (default of TRUE) of whether to revise the estimates and standard erros for binary independant variables. If FALSE, derivatives are taken on binary independant variables as continuous variables.
`digits`	number of digits for output.
`...`	additional arguments to be passed.

Details

Marginal effects from a binary probit or logit model is calculated. The two choices are the method of averaging effects and revising estimates for dummy variables. Marginal effects can be calculated at the mean of the independent variables (i.e., `x.mean = TRUE`), or as the average of individual marginal effects at each observation (i.e., `x.mean = FALSE`). `rev.dum = TRUE` allows marginal effects for dummy variables are calculated differently, instead of treating them as continuous variables.

Value

Return a list object of class "maBina" with the following components:

- `link` — link function used in the binary model;
- `f.xb` — scale factor of marginal effects, calculated as the density function evaluated at the means of the variables when `x.mean = TRUE` is specified or the average density value for all individual observations when `x.mean = FALSE` is specified;
- `w` — a binary probit or logit model object estimated from `glm()`;
- `out` — a data frame object of marginal effects, t-value, and p-value.

Author(s)

Changyou Sun (<csun@cfr.msstate.edu>)

References

Greene, W.H. 2003. Econometric Analysis (5th ed.). Prentice Hall, New York. 1026 P.

See Also

maTrend; plot.maTrend.

Examples

```
data(daPe)
ma <- glm(grade ~ gpa + tuce + psi, x = TRUE,
    data = daPe, family = binomial(link = "probit"))
ea <- maBina(w = ma, x.mean = TRUE, rev.dum = TRUE)
ea
```

`maTrend` *Trend of Marginal Effects*

Description

This function computes the change of probability for a continuous variable, and furthermore, stratifies the probability through a binary independant variable.

Usage

```
maTrend(q, n = 300, nam.c, nam.d, ...)
```

Arguments

- `q` — a object of class of "maBina" estimated from `maBina()`.
- `n` — number of points for calculating probability; the large the number, the smoother the curve.
- `nam.c` — a name of a continuous indepedent variable; this must be given for the function to work.
- `nam.d` — an optional name of a binary independent variable; this is used to stratify the probability.
- `...` — additional arguments to be passed.

Details

Marginal effects are calcuated at each value of a continous variable. If specficied, the trend can be stratified by a binary independent variable.

Value

Return a list object of class "maTrend" with the following components:

`q`	a list object of class "maBina";
`nam.c`	the name of a continous variable;
`mm`	matrix of independant variables for all
`trend`	a data frame of the continous variable and probability values; if `nam.d` is specificied, the data frame also contains the probability values stratified by the dummy variable;
`nam.d`	if `nam.d` is specified, the name of a binary variable .
`m1`	if `nam.d` is specified, the matrix of `mm` with the column value for `nam.d` replaced by 1
`m0`	if `nam.d` is specified, the matrix of `mm` with the column value for `nam.d` replaced by 0

Author(s)

Changyou Sun (<csun@cfr.msstate.edu>)

References

Greene, W.H. 2003. Econometric Analysis (5th ed.). Prentice Hall, New York. 1026 P.

See Also

maBina; print.maTrend; plot.maTrend.

Examples

```
data(daPe)
ma <- glm(grade ~ gpa + tuce + psi, x = TRUE,
    data = daPe, family = binomial(link = "probit"))
summary(ma)

(ea <- maBina(w = ma, x.mean = TRUE, rev.dum = TRUE))
(ta <- maTrend(q = ea, nam.c = "gpa", nam.d= "psi"))
plot(ta)
```

`plot.evReturn` *Plot for Average Cumulative Abnormal Returns from Event Analysis*

Description

Plot average cumulative abnormal returns from event analysis versus days in event window.

Usage

```
## S3 method for class 'evReturn'
plot(x, ...)
```

Arguments

`x`	an object of class "evReturn".
`...`	additional arguments to be passed.

Details

Plot average cumulative abnormal returns from event analysis versus days in event window. This is for all firms as a group and is called HNt in Mei and Sun (2008).

Author(s)

Changyou Sun (`<csun@cfr.msstate.edu>`)

References

Mei, B., and C. Sun. 2008. Event analysis of the impact of mergers and acquisitions on the financial performance of the U.S. forest products industry. Forest Policy and Economics 10(5):286-294.

See Also

`evReturn`; `print.evReturn`.

Examples

```
# see Mei and Sun (2008).
```

`plot.maTrend` *Plot for Marginal Effect Trends*

Description

Plot the probability values versus a continous variable with a stratification by a dummy variable.

Usage

```
## S3 method for class 'maTrend'
plot(x, ...)
```

Arguments

`x`	an object of class "maTrend".
`...`	additional arguments to be passed.

Details

Plot the probability values for a continuous variable. If a strata is specified through `nam.d` in `maTrend()`, then the stratified values also are shown.

Value

A plot of probability values

Author(s)

Changyou Sun (<`csun@cfr.msstate.edu`>)

References

Greene, W.H. 2003. Econometric Analysis (5th ed.). Prentice Hall, New York. 1026 P.

See Also

`maTrend`; `print.maTrend`.

Examples

```
data(daPe)
ma <- glm(grade ~ gpa + tuce + psi, x = TRUE,
    data = daPe, family = binomial(link = "probit"))
ea <- maBina(w = ma, x.mean = TRUE, rev.dum = TRUE)
ta <- maTrend(q = ea, nam.c = "gpa", nam.d= "psi")
plot(ta)
```

print.aiFit — *Printing results from AIDS models*

Description

Show estimation resutls from static or dynamic AIDS models estimated from `aiStaFit`, `aiStaHau`, and `aiDynFit`.

Usage

```
## S3 method for class 'aiFit'
print(x, ...)
```

Arguments

`x`	an object of class `aiFit` from the function of `aiStaFit`, `aiStaHau`, or `aiDynFit`.
`...`	additional arguments to be passed.

Details

This print method for object of class `aiFit`.

Value

Summary results of the coefficents or outputs.

Author(s)

Changyou Sun (<csun@cfr.msstate.edu>)

See Also

`aiStaFit`; `aiStaHau`; `aiDynFit`.

Examples

```
# see the examples for 'aiDynFit'.
```

print.evReturn — *Printing Abnormal Return Results from Event Analsyis*

Description

Show abnormal return resutls from `evReturn`.

Usage

```
## S3 method for class 'evReturn'
print(x, ...)
```

Arguments

`x`	an object of class `evReturn`.
`...`	additional arguments to be passed.

Details

This print method for object of class `evReturn`.

Value

Summary results of the coefficents or outputs.

Author(s)

Changyou Sun (<csun@cfr.msstate.edu>)

See Also

`evReturn`; `plot.evReturn`.

Examples

```
# see the examples for 'evReturn'.
```

`print.evRisk` *Printing Risk Results from Event Analsyis*

Description

Show regression resutls from `evRisk`.

Usage

```
## S3 method for class 'evRisk'
print(x, ...)
```

Arguments

`x`	an object of class `evRisk`.
`...`	additional arguments to be passed.

Details

This print method for object of class `evRisk`.

Value

Summary results of the coefficents or outputs.

Author(s)

Changyou Sun (<csun@cfr.msstate.edu>)

See Also

evRisk.

Examples

```
# see the examples for 'evRisk'.
```

---

print.maTrend *Printing Marginal Effect Trends*

Description

Show dimension and some values of probability values from maTrend.

Usage

```
## S3 method for class 'maTrend'
print(x, ...)
```

Arguments

x	an object of class maTrend from the function of maTrend.
...	additional arguments to be passed.

Details

This print method for maTrend shows the probability values.

Value

Summary results of the probability value estimates.

Author(s)

Changyou Sun (<csun@cfr.msstate.edu>)

See Also

maTrend; plot.maTrend.

Examples

```
data(daPe)
ma <- glm(grade ~ gpa + tuce + psi, x = TRUE,
    data = daPe, family = binomial(link = "probit"))
ea <- maBina(w = ma, x.mean = TRUE, rev.dum = TRUE)
ta <- maTrend(q = ea, nam.c = "gpa", nam.d= "psi")
ta
```

`summary.aiFit` *Summary of Results from Static or Dynamic Models*

Description

This summarizes the main results from AIDS models.

Usage

```
## S3 method for class 'aiFit'
summary(object, digits=3, ...)
```

Arguments

`object`	an object of class `aiFit` from the function of `aiStaFit` or `aiDynFit`.
`digits`	number of digits for rounding outputs
`...`	additional arguments to be passed.

Details

This wraps up the coefficents and statistics from aiFit by equation.

Value

A data frame object with coefficients and related statistics by equation.

Author(s)

Changyou Sun (<csun@cfr.msstate.edu>)

See Also

`aiStaFit` and `aiDynFit`.

Examples

```
# see the examples for 'aiDynFit'.
```

Index