

Package ‘Zelig’

April 17, 2009

Version 3.4-5

Date 2009-03-13

Title Everyone’s Statistical Software

Author Kosuke Imai <kimai@Princeton.Edu>, Gary King <king@harvard.edu>, Olivia Lau <olau@fas.harvard.edu>

Maintainer Kosuke Imai <kimai@Princeton.Edu>

Depends R (>= 2.6.0), MASS, boot

Description Zelig is an easy-to-use program that can estimate, and help interpret the results of, an enormous range of statistical models. It literally is “everyone’s statistical software” because Zelig’s simple unified framework incorporates everyone else’s (R) code. We also hope it will become “everyone’s statistical software” for applications and teaching, and so have designed Zelig so that anyone can easily use it or add their programs to it. Zelig also comes with infrastructure that facilitates the use of any existing method, such as by allowing multiply imputed data for any model, and mimicking the program Clarify (for Stata) that takes the raw output of existing statistical procedures and translates them into quantities of direct interest.

License GPL (>= 2)

URL <http://gking.harvard.edu/zelig>

Suggests VGAM (>= 0.7-5), MCMCpack (>= 0.8-2), mvtnorm, survival, sandwich (>= 2.1-0), zoo (>= 1.5-0), coda, nnet, sna, gee, systemfit, mgcv, lme4, anchors (>= 2.0), survey

Repository CRAN

Date/Publication 2009-04-04 20:39:10

R topics documented:

Zelig-package	3
approval	6
bivariate	6
coalition	7

coalition2	8
current.packages	8
dims	9
eidat	10
free1	10
free2	12
friendship	13
grunfeld	14
gsource	15
help.zelig	16
hoff	16
homerun	17
immigration	18
klein	18
kmenta	19
macro	19
match.data	20
MatchIt.url	20
mexico	21
mi	21
mid	22
model.end	23
model.frame.multiple	24
model.matrix.multiple	25
network	26
newpainters	27
parse.formula	28
parse.par	30
PERisk	31
plot.ci	32
plot.surv	34
plot.zelig	35
put.start	36
repl	37
rocplot	38
sanction	40
set.start	40
setx	41
sim	43
sna.ex	45
summary.zelig	46
SupremeCourt	47
swiss	47
ternaryplot	48
ternarypoints	50
tobin	51
turnout	51
user.prompt	52

voteincome	52
Weimar	53
zelig	54
Zelig.url	55
zeligDepStatus	55
zeligDepUpdate	57
zeligDescribeModelXML	58

Index	60
--------------	-----------

Zelig-package *Zelig: Everyone's Statistical Software*

Description

Zelig is an easy-to-use program that can estimate, and help interpret the results of, an enormous range of statistical models. It literally is “everyone’s statistical software” because Zelig’s simple unified framework incorporates everyone else’s (R) code. We also hope it will become “everyone’s statistical software” for applications and teaching, and so have designed Zelig so that anyone can easily use it or add their programs to it. Zelig also comes with infrastructure that facilitates the use of any existing method, such as by allowing multiply imputed data for any model, and mimicking the program Clarify (for Stata) that takes the raw output of existing statistical procedures and translates them into quantities of direct interest.

Details

Package: Zelig
Version: 2.8-5
Date: 2007-06-12
Depends: R (>= 2.4.0), MASS, boot
Suggests: VGAM (>= 0.7-1), MCMCpack (>= 0.7-4), mvtnorm, survival, sandwich (>= 2.0-0), zoo (>= 1.2-1), coda, nnet, s
License: GPL version 2 or newer
URL: <http://gking.harvard.edu/zelig>

Index:

approval	U.S. Presidential Approval Data
coalition	Coalition Dissolution in Parliamentary Democracies
current.packages	Find all packages in a dependency chain
dims	Return Dimensions of Vectors, Arrays, and Data Frames
eidat	Simulation Data for Ecological Inference
friendship	Simulated Example of Schoolchildren Friendship Network
gsource	Read Data As a Space-Delimited Table
help.zelig	HTML Help for Zelig Commands and Models

hoff	Social Security Expenditure Data
immigration	Individual Preferences Over Immigration Policy
macro	Macroeconomic Data
match.data	Output matched data sets
MatchIt.url	Table of links for Zelig
mexico	Voting Data from the 1988 Mexican Presidential Election
mi	Bundle multiply imputed data sets as a list
mid	Militarized Interstate Disputes
model.end	Cleaning up after optimization
model.frame.multiple	Extracting the "environment" of a model formula
model.matrix.multiple	Design matrix for multivariate models
network	Format matrices into a data frame for social network analysis
newpainters	The Discretized Painter's Data of de Piles
parse.formula	Parsing user-input formulas into multiple syntax
parse.par	Select and reshape parameter vectors
PERisk	Political Economic Risk Data from 62 Countries in 1987
plot.ci	Plotting Vertical confidence Intervals
plot.zelig	Graphing Quantities of Interest
put.start	Set specific starting values for certain parameters
repl	Replicating Analyses
rocplot	Receiver Operator Characteristic Plots
sanction	Multilateral Economic Sanctions
set.start	Set starting values for all parameters
setx	Setting Explanatory Variable Values
sim	Simulating Quantities of Interest
sna.ex	Simulated Example of Social Network Data
summary.zelig	Summary of Simulated Quantities of Interest
SupremeCourt	U.S. Supreme Court Vote Matrix
swiss	Swiss Fertility and Socioeconomic Indicators (1888) Data
ternaryplot	Ternary diagram
ternarypoints	Adding Points to Ternary Diagrams
tobin	Tobin's Tobit Data
turnout	Turnout Data Set from the National Election Survey
user.prompt	Pause in demo files
Weimar	1932 Weimar election data
zelig	Estimating a Statistical Model
zeligDepStatus	Zelig Dependencies Packages Client Status
zeligDepUpdate	Download Zelig Dependencies Packages
zeligDescribeModelXML	Zelig interface functions
Zelig-package	Everyone's Statistical Software
Zelig.url	Table of links for Zelig

tobit	Linear regression for Left-Censored Dependet Variable (source)
tobit.bayes	Bayesian Linear Regression for a Censored Dependent Variable (source)
twosls	Two Stage Least Squares (source)
weibull	Weibull Regression for Duration Dependent Variables (source)

Author(s)

Kosuke Imai <kimai@Princeton.Edu>, Gary King <king@harvard.edu>, Olivia Lau <olau@fas.harvard.edu>
 Maintainer: Kosuke Imai <kimai@Princeton.Edu>

approval	<i>U.S. Presidential Approval Data</i>
----------	--

Description

Monthly public opinion data for 2001-2006.

Usage

data(approval)

Format

A table containing 8 variables ("month", "year", "approve", "disapprove", "unsure", "sept.oct.2001", "iraq.war", and "avg.price") and 65 observations.

Source

ICPSR

References

Stuff here

bivariate	<i>Sample data for bivariate probit regression</i>
-----------	--

Description

Sample data for the bivariate probit regression.

Usage

data(bivariate)

Format

A table containing 6 variables ("y1", "y2", "x1", "x2", "x3", and "x4") and 78 observations.

Source

This is a cleaned and relabelled version of the sanction data set, available in Zelig.

References

Martin, Lisa (1992). *Coercive Cooperation: Explaining Multilateral Economic Sanctions*, Princeton: Princeton University Press.

coalition

Coalition Dissolution in Parliamentary Democracies

Description

This data set contains survival data on government coalitions in parliamentary democracies (Belgium, Canada, Denmark, Finland, France, Iceland, Ireland, Israel, Italy, Netherlands, Norway, Portugal, Spain, Sweden, and the United Kingdom) for the period 1945-1987. For parsimony, country indicator variables are omitted in the sample data.

Usage

```
data(coalition)
```

Format

A table containing 7 variables ("duration", "ciep12", "invest", "fract", "polar", "numst2", "crisis") and 314 observations. For variable descriptions, please refer to King, Alt, Burns and Laver (1990).

Source

ICPSR

References

King, Gary, James E. Alt, Nancy Elizabeth Burns and Michael Laver (1990). "A Unified Model of Cabinet Dissolution in Parliamentary Democracies," *American Journal of Political Science*, vol. 34, no. 3, pp. 846-870.

Gary King, James E. Alt, Nancy Burns, and Michael Laver. ICPSR Publication Related Archive, 1115.

coalition2

Coalition Dissolution in Parliamentary Democracies, Modified Version

Description

This data set contains survival data on government coalitions in parliamentary democracies (Belgium, Canada, Denmark, Finland, France, Iceland, Ireland, Israel, Italy, Netherlands, Norway, Portugal, Spain, Sweden, and the United Kingdom) for the period 1945-1987. Country indicator variables are included in the sample data.

Usage

```
data(coalition2)
```

Format

A data frame containing 8 variables ("duration", "ciep12", "invest", "fract", "polar", "numst2", "crisis", "country") and 314 observations. For variable descriptions, please refer to King, Alt, Burns and Laver (1990).

Source

ICPSR

References

King, Gary, James E. Alt, Nancy Elizabeth Burns and Michael Laver (1990). "A Unified Model of Cabinet Dissolution in Parliamentary Democracies," *American Journal of Political Science*, vol. 34, no. 3, pp. 846-870.

Gary King, James E. Alt, Nancy Burns, and Michael Laver. ICPSR Publication Related Archive, 1115.

current.packages*Find all packages in a dependency chain*

Description

Use `current.packages` to find all the packages suggested or required by a given package, and the currently installed version number for each.

Usage

```
current.packages(package)
```

Arguments

`package` a character string corresponding to the name of an installed package

Value

A matrix containing the current version number of the packages suggested or required by `package`.

Author(s)

Olivia Lau <olau@fas.harvard.edu>

See Also

`packageDescription`

Examples

```
## Not run:  
current.packages("Zelig")  
## End(Not run)
```

dims

Return Dimensions of Vectors, Arrays, and Data Frames

Description

Retrieve the dimensions of a vector, array, or data frame.

Usage

```
dims(x)
```

Arguments

`x` An R object. For example, a vector, matrix, array, or data frame.

Value

The function `dims` performs exactly the same as `dim`, and additionally returns the `length` of vectors (treating them as one-dimensional arrays).

Author(s)

Olivia Lau <olau@fas.harvard.edu>

See Also

`dim`, `length`

Examples

```
a <- 1:12
dims(a)

a <- matrix(1, nrow = 4, ncol = 9)
dims(a)
```

eidat	<i>Simulation Data for Ecological Inference</i>
-------	---

Description

This dataframe contains a simulated data set to illustrate the models for ecological inference.

Usage

```
data(eidat)
```

Format

A table containing 4 variables ("t0", "t1", "x0", "x1") and 10 observations.

freel	<i>Freedom of Speech Data</i>
-------	-------------------------------

Description

Selection of individual-level survey data for freedom of speech.

Usage

```
data(freel)
```

Details

A table with 150 observations and 12 variables.

sex 1 for men and 0 for women

age Age of respondent in years

educ Levels of education, coded as a numeric variable with

- 1 No formal education
- 2 Less than primary school education
- 3 Completed primary school
- 4 Completed secondary school
- 5 Completed high school

- 6 Completed college
- 7 Completed post-graduate degree

country Character strings consisting of "Oceania", "Eurasia", and "Eastasia", after Orwell's *1984*.

y Self assessment (see below).

v1-v6 Response to vignettes (see below).

Survey respondents were asked in almost the same language for a self-assessment and for an assessment of several hypothetical persons described by written vignettes. The self assessment (`self`, in the data set), "How free do you think [name/you] [is/are] to express [him-her/your]self without fear of government reprisal?" was first asked of the survey respondent with respect to him or herself, and then after each of vignette. The possible response categories are:

- 1 Completely free
- 2 Very free
- 3 Moderately free
- 4 Slightly free
- 5 Not free at all

The vignettes, ordered from most free to least free, are:

- vign1 [Kay] does not like many of the government's policies. She frequently publishes her opinion in newspapers, criticizing decisions by officials and calling for change. She sees little reason these actions could lead to government reprisal.
- vign2 [Michael] disagrees with many of the government's policies. Though he knows criticism is frowned upon, he doesn't believe the government would punish someone for expressing critical views. He makes his opinion known on most issues without regard to who is listening.
- vign3 [Bob] has political views at odds with the government. He has heard of people occasionally being arrested for speaking out against the government, and government leaders sometimes make political speeches condemning those who criticize. He sometimes writes letters to newspapers about politics, but he is careful not to use his real name.
- vign4 [Connie] does not like the government's stance on many issues. She has a friend who was arrested for being too openly critical of governmental leaders, and so she avoids voicing her opinions in public places.
- vign5 [Vito] disagrees with many of the government's policies, and is very careful about whom he says this to, reserving his real opinions for family and close friends only. He knows several men who have been taken away by government officials for saying negative things in public.
- vign6 [Sonny] lives in fear of being harassed for his political views. Everyone he knows who has spoken out against the government has been arrested or taken away. He never says a word about anything the government does, not even when he is at home alone with his family.

References

WHO's World Health Survey by Lydia Bendib, Somnath Chatterji, Alena Petrakova, Ritu Sadana, Joshua A. Salomon, Margie Schneider, Bedirhan Ustun, Maria Villanueva
Jonathan Wand, Gary King and Olivia Lau. (2007) "Anchors: Software for Anchoring Vignettes". *Journal of Statistical Software*. Forthcoming. copy at <http://wand.stanford.edu/research/anchors-jss.pdf>

Gary King and Jonathan Wand. "Comparing Incomparable Survey Responses: New Tools for Anchoring Vignettes," *Political Analysis*, 15, 1 (Winter, 2007): Pp. 46-66, copy at <http://gking.harvard.edu/files/abs/c-abs.shtml>.

free2

Freedom of Speech Data

Description

Selection of individual-level survey data for freedom of speech.

Usage

`data(free2)`

Details

A table with 150 observations and 12 variables.

`sex` 1 for men and 0 for women

`age` Age of respondent in years

`educ` Levels of education, coded as a numeric variable with

- 1 No formal education
- 2 Less than primary school education
- 3 Completed primary school
- 4 Completed secondary school
- 5 Completed high school
- 6 Completed college
- 7 Completed post-graduate degree

`country` Character strings consisting of "Oceania", "Eurasia", and "Eastasia", after Orwell's *1984*.

`y` Self assessment (see below).

`v1-v6` Response to vignettes (see below).

Survey respondents were asked in almost the same language for a self-assessment and for an assessment of several hypothetical persons described by written vignettes. The self assessment (`self`, in the data set), "How free do you think [name/you] [is/are] to express [him-her/your]self without fear of government reprisal?" was first asked of the survey respondent with respect to him or herself, and then after each of vignette. The possible response categories are:

- 1 Completely free
- 2 Very free
- 3 Moderately free
- 4 Slightly free
- 5 Not free at all

The vignettes, ordered from most free to least free, are:

- vign1 [Kay] does not like many of the government's policies. She frequently publishes her opinion in newspapers, criticizing decisions by officials and calling for change. She sees little reason these actions could lead to government reprisal.
- vign2 [Michael] disagrees with many of the government's policies. Though he knows criticism is frowned upon, he doesn't believe the government would punish someone for expressing critical views. He makes his opinion known on most issues without regard to who is listening.
- vign3 [Bob] has political views at odds with the government. He has heard of people occasionally being arrested for speaking out against the government, and government leaders sometimes make political speeches condemning those who criticize. He sometimes writes letters to newspapers about politics, but he is careful not to use his real name.
- vign4 [Connie] does not like the government's stance on many issues. She has a friend who was arrested for being too openly critical of governmental leaders, and so she avoids voicing her opinions in public places.
- vign5 [Vito] disagrees with many of the government's policies, and is very careful about whom he says this to, reserving his real opinions for family and close friends only. He knows several men who have been taken away by government officials for saying negative things in public.
- vign6 [Sonny] lives in fear of being harassed for his political views. Everyone he knows who has spoken out against the government has been arrested or taken away. He never says a word about anything the government does, not even when he is at home alone with his family.

References

WHO's World Health Survey by Lydia Bendib, Somnath Chatterji, Alena Petrakova, Ritu Sadana, Joshua A. Salomon, Margie Schneider, Bedirhan Ustun, Maria Villanueva

Jonathan Wand, Gary King and Olivia Lau. (2007) "Anchors: Software for Anchoring Vignettes". *Journal of Statistical Software*. Forthcoming. copy at <http://wand.stanford.edu/research/anchors-jss.pdf>

Gary King and Jonathan Wand. "Comparing Incomparable Survey Responses: New Tools for Anchoring Vignettes," *Political Analysis*, 15, 1 (Winter, 2007): Pp. 46-66, copy at <http://gking.harvard.edu/files/abs/c-abs.shtml>.

friendship

Simulated Example of Schoolchildren Friendship Network

Description

This data set contains six sociomatrices of simulated data on friendship ties among schoolchildren.

Usage

`data(friendship)`

Format

Each variable in the dataset is a 15 by 15 matrix representing some form of social network tie held by the fictitious children. The matrices are labeled "friends", "advice", "prestige", "authority", "perpower" and "per".

The sociomatrices were combined into the friendship dataset using the `format.network.data` function from the `netglm` package by Skyler Cranmer as shown in the example.

Source

fictitious

Examples

```
## Not run:
      friendship <- format.network.data(friends, advice, prestige, authority, perpower)
## End(Not run)
```

grunfeld

Simulation Data for model Seemingly Unrelated Regression (sur) that corresponds to method SUR of systemfit

Description

Dataframe contains 20 annual observations from 1935 to 1954 of 7 variables for two firms General Electric and Westinghouse. Columns are Year; Ige and Iw = Gross investment for GE and W, respectively; Fge and Fw=Market value of Firm as of begin of the year; Cge and Cw= Capital stock measure as of begin of the year.

Usage

```
data(grunfeld)
```

Format

A table containing 7 variables ("Year", "Ige", "Fge", "Cge", "Iw", "Fw", "Cw") and 20 observations.

`gsource`*Read Data As a Space-Delimited Table*

Description

The `gsource` function allows you to read a space delimited table as a data frame. Unlike `scan`, you may use `gsource` in a sourced file, and unlike `read.table`, you may use `gsource` to include a small (or large) data set in a file that also contains other commands.

Usage

```
gsource(var.names = NULL, variables)
```

Arguments

`var.names` An optional vector of character strings representing the column names. By default, `var.names = NULL`.

`variables` A single character string representing the data.

Value

The output from `gsource` is a data frame, which you may save to an object in your workspace.

Author(s)

Olivia Lau <olau@fas.harvard.edu>

See Also

`read.table`, `scan`

Examples

```
## Not run:
data <- gsource(variables = "
  1 2 3 4 5
  6 7 8 9 10
  3 4 5 1 3
  6 7 8 1 9  ")

data <- gsource(var.names = "Vote Age Party", variables = "
  0 23 Democrat
  0 27 Democrat
  1 45 Republican
  1 65 Democrat  ")

## End(Not run)
```

`help.zelig`*HTML Help for Zelig Commands and Models*

Description

The `help.zelig` command launches html help for Zelig commands and supported models. The full manual is available online at <http://gking.harvard.edu/zelig>.

Usage

```
help.zelig(...)
```

Arguments

... a Zelig command or model. `help.zelig(command)` will take you to an index of Zelig commands and `help.zelig(model)` will take you to a list of models.

Author(s)

Kosuke Imai <kimai@princeton.edu>; Gary King <king@harvard.edu>; Olivia Lau <olau@fas.harvard.edu>

See Also

The complete document is available online at <http://gking.harvard.edu/zelig>.

`hoff`*Social Security Expenditure Data*

Description

This data set contains annual social security expenditure (as percent of budget lagged by two years), the relative frequency of mentions social justice received in the party's platform in each year, and whether the president is Republican or Democrat.

Usage

```
data(hoff)
```

Format

A table containing 5 variables ("year", "L2SocSec", "Just503D", "Just503R", "RGovDumy") and 36 observations.

Source

ICPSR (replication dataset s1109)

References

Gary King and Michael Laver. "On Party Platforms, Mandates, and Government Spending," *American Political Science Review*, Vol. 87, No. 3 (September, 1993): pp. 744-750.

homerun

Sample Data on Home Runs Hit By Mark McGwire and Sammy Sosa in 1998.

Description

Game-by-game information for the 1998 season for Mark McGwire and Sammy Sosa. Data are a subset of the dataset provided in Simonoff (1998).

Usage

```
data(homerun)
```

Format

A data frame containing 5 variables ("gameno", "month", "homeruns", "playerstatus", "player") and 326 observations.

gameno an integer variable denoting the game number

month a factor variable taking with levels "March" through "September" denoting the month of the game

homeruns an integer vector denoting the number of homeruns hit in that game for that player

playerstatus an integer vector equal to "0" if the player played in the game, and "1" if they did not.

player an integer vector equal to "0" (McGwire) or "1" (Sosa)

Source

<http://www.amstat.org>

References

Simonoff, Jeffrey S. 1998. "Move Over, Roger Maris: Breaking Baseball's Most Famous Record." *Journal of Statistics Education* 6(3). Data used are a subset of the data in the article.

 immigration

Individual Preferences Over Immigration Policy

Description

These five datasets are part of a larger set of 10 multiply imputed data sets describing individual preferences toward immigration policy. Imputation was performed via Amelia.

Format

Each multiply-imputed data set consists of a table with 7 variables ("ipip", "wage1992", "prtyid", "ideol", "gender") and 2,485 observations. For variable descriptions, please refer to Scheve and Slaughter, 2001.

Source

National Election Survey

References

Scheve, Kenneth and Matthew Slaughter (2001). "Labor Market Competition and Individual Preferences Over Immigration Policy," *The Review of Economics and Statistics*, vol. 83, no. 1, pp. 133-145.

 klein

Simulation Data for model Two-Stage Least Square (twosls) that corresponds to method 2SLS of systemfit

Description

Dataframe contains annual observations of US economy from 1920 to 1940. The columns are, Year, C=Consumption, P=Corporate profits, P1=Previous year corporate profit, Wtot=Total wage, Wp=Private wage bill, Wg=Government wage bill, I=Investment, K1=Previous year capital stock, X=GNP, G=Government spending, T=Taxes, X1=Previous year GNP, Tm=Year-1931.

Usage

```
data(klein)
```

Format

A table containing 14 variables ("year", "C", "P", "P1", "Wtot", "Wp", "Wg", "I", "K1", "X", "G", "T", "X1", "Tm") and 21 observations.

Source

<http://pages.stern.nyu.edu/~wgreene/Text/econometricanalysis.htm>

kmenta	<i>Simulation Data for model Three-Stage Least Square (threesls) that corresponds to method 3SLS of systemfit</i>
--------	---

Description

Dataframe contains 20 annual observations of a supply/demand model with 5 variables. Columns are q=Food consumption per capita, p=Ratio of food price to general consumer prices, d=Disposable income in constant dollars, f=Ratio of preceding year's prices received by farmers to general consumer prices, a=Time index.

Usage

```
data(kmenta)
```

Format

A table containing 5 variables ("q", "p", "d", "f", "a") and 20 observations.

macro	<i>Macroeconomic Data</i>
-------	---------------------------

Description

Selected macroeconomic indicators for Austria, Belgium, Canada, Denmark, Finland, France, Italy, Japan, the Netherlands, Norway, Sweden, the United Kingdom, the United States, and West Germany for the period 1966-1990.

Usage

```
data(macro)
```

Format

A table containing 6 variables ("country", "year", "gdp", "unem", "capmob", and "trade") and 350 observations.

Source

ICPSR

References

King, Gary, Michael Tomz and Jason Wittenberg. ICPSR Publication Related Archive, 1225.
 King, Gary, Michael Tomz and Jason Wittenberg (2000). "Making the Most of Statistical Analyses: Improving Interpretation and Presentation," *American Journal of Political Science*, vol. 44, pp. 341-355.

<code>match.data</code>	<i>Output matched data sets</i>
-------------------------	---------------------------------

Description

The code `match.data` creates output data sets from the `matchit` matching algorithm.

Usage

```
match.data <- match.data(object, group = "all")
```

Arguments

<code>object</code>	Stored output from <code>matchit</code> .
<code>group</code>	Which units to output. Selecting "all" (default) gives all matched units (treated and control), "treat" gives just the matched treated units, and "control" gives just the matched control units.

Value

The `match.data` command generates a matched data set from the output of the `matchit` function, according to the options selected in the `group` argument. The matched data set contains the additional variables:

<code>pscore</code>	The propensity score for each unit.
<code>psclass</code>	The subclass index for each unit (if applicable).
<code>psweights</code>	The weight for each unit (generated from the matching procedure).

See the `matchit` documentation for more details on these items.

Author(s)

Daniel Ho <deho@fas.harvard.edu>; Kosuke Imai <kimai@princeton.edu>; Gary King <king@harvard.edu>; Elizabeth Stuart <stuart@stat.harvard.edu>

See Also

The complete documentation for `matchit` is available online at <http://gking.harvard.edu/matchit>.

<code>MatchIt.url</code>	<i>Table of links for Zelig</i>
--------------------------	---------------------------------

Description

Table of links for `help.zelig` for the companion `MatchIt` package.

`mexico`*Voting Data from the 1988 Mexican Presidential Election*

Description

This dataset contains voting data for the 1988 Mexican presidential election.

Usage

```
data(mexico)
```

Format

A table containing 33 variables and 1,359 observations.

Source

ICPSR

References

King, Gary, Michael Tomz and Jason Wittenberg (2000). "Making the Most of Statistical Analyses: Improving Interpretation and Presentation," *American Journal of Political Science*, vol. 44, pp. 341-355.

King, Tomz and Wittenberg. ICPSR Publication Related Archive, 1255.

`mi`*Bundle multiply imputed data sets as a list*

Description

The code `mi` bundles multiply imputed data sets as a list for further analysis.

Usage

```
mi(...)
```

Arguments

`...` multiply imputed data sets, separated by commas. The arguments can be tagged by `name=data` where `name` is the element named used for the data set `data`.

Value

The list containing each multiply imputed data set as an element. The class name is `mi`. The list can be inputted into `zelig` for statistical analysis with multiply imputed data sets. See `zelig` for details.

Author(s)

Kosuke Imai <<kimai@princeton.edu>>; Gary King <<king@harvard.edu>>; Olivia Lau <<olau@fas.harvard.edu>>

See Also

The full Zelig manual is available at <http://gking.harvard.edu/zelig>.

Examples

```
data(immi1, immi2, immi3, immi4, immi5)
mi(immi1, immi2, immi3, immi4, immi5)
```

mid

Militarized Interstate Disputes

Description

A small sample from the militarized interstate disputes database, available at http://pss.la.psu.edu/MID_DATA.HTM.

Usage

```
data(mid)
```

Format

A table containing 6 variables ("conflict", "major", "contig", "power", "maxdem", "mindem", and "years") and 3,126 observations. For full variable descriptions, please see King and Zeng, 2001.

Source

Militarized Interstate Disputes database

References

King, Gary, and Lanche Zeng (2001). "Explaining Rare Events in International Relations," *International Organization*, vol. 55, no. 3, pp. 693-715.

Jones, Daniel M., Stuart A. Bremer and David Singer (1996). "Militarized Interstate Disputes, 1816-1992: Rationale, Coding Rules, and Empirical Patterns," *Conflict Management and Peace Science*, vol. 15, no. 2, pp. 163-213.

model.end	<i>Cleaning up after optimization</i>
-----------	---------------------------------------

Description

The `model.end` function creates a list of regression output from `optim` output. The list includes coefficients (from the `optim` `par` output), a variance-covariance matrix (from the `optim` Hessian output), and any terms, contrasts, or xlevels (from the model frame). Use `model.end` after calling `optim`, but before assigning a class to the regression output.

Usage

```
model.end(res, mf)
```

Arguments

<code>res</code>	the output from <code>optim</code> or another fitting-algorithm
<code>mf</code>	the model frame output by <code>model.frame</code>

Value

A list of regression output, including:

<code>coefficients</code>	the optimized parameters
<code>variance</code>	the variance-covariance matrix (the negative inverse of the Hessian matrix returned from the optimization procedure)
<code>terms</code>	the terms object. See <code>terms.object</code> for more information
<code>...</code>	additional elements passed from <code>res</code>

Author(s)

Kosuke Imai <kimai@princeton.edu>; Gary King <king@harvard.edu>; Olivia Lau <olau@fas.harvard.edu>; Ferdinand Alimadhi <falimadhi@iq.harvard.edu>

See Also

The full Zelig manual at <http://gking.harvard.edu/zelig> for examples.

`model.frame.multiple`*Extracting the “environment” of a model formula*

Description

Use `model.frame.multiple` after `parse.par` to create a data frame of the unique variables identified in the formula (or list of formulas).

Usage

```
model.frame.multiple(formula, data, eqn = NULL, ...)
```

Arguments

<code>formula</code>	a list of formulas of class "multiple", returned from <code>parse.par</code>
<code>data</code>	a data frame containing all the variables used in <code>formula</code>
<code>eqn</code>	an optional character string or vector of character strings specifying the equations (specified in <code>describe.mymodel</code>) for which you would like to pull out the relevant variables.
<code>...</code>	additional arguments passed to <code>model.frame.default</code>

Value

The output is a data frame (with a `terms` attribute) containing all the unique explanatory and response variables identified in the list of formulas. By default, missing (NA) values are listwise deleted.

If `as.factor` appears on the left-hand side, the response variables will be returned as an indicator (0/1) matrix with columns corresponding to the unique levels in the factor variable.

If any formula contains more than one `tag` statement, `model.frame.multiple` will return the original variable in the data frame and use the `tag` information in the `terms` attribute only.

Author(s)

Kosuke Imai <kimai@princeton.edu>; Gary King <king@harvard.edu>; Olivia Lau <olau@fas.harvard.edu>; Ferdinand Alimadhi <falimadhi@iq.harvard.edu>

See Also

`model.matrix.default`, `parse.formula` and the full Zelig manual at <http://gking.harvard.edu/zelig>

Examples

```
## Not run:
data(sanction)
formulae <- list(import ~ coop + cost + target,
                 export ~ coop + cost + target)
fml <- parse.formula(formulae, model = "bivariate.logit")
D <- model.frame(fml, data = sanction)
## End(Not run)
```

```
model.matrix.multiple
```

Design matrix for multivariate models

Description

Use `model.matrix.multiple` after `parse.formula` to create a design matrix for multiple-equation models.

Usage

```
model.matrix.multiple(object, data, shape = "compact", eqn = NULL, ...)
```

Arguments

<code>object</code>	the list of formulas output from <code>parse.formula</code>
<code>data</code>	a data frame created with <code>model.frame.multiple</code>
<code>shape</code>	a character string specifying the shape of the outputted matrix. Available options are <ul style="list-style-type: none"> "compact" (default) the output matrix will be an $n \times v$, where v is the number of unique variables in all of the equations (including the intercept term) "array" the output is an $n \times K \times J$ array where J is the total number of equations and K is the total number of parameters across all the equations. If a variable is not in a certain equation, it is observed as a vector of 0s. "stacked" the output will be a $2n \times K$ matrix where K is the total number of parameters across all the equations.
<code>eqn</code>	a character string or a vector of character strings identifying the equations from which to construct the design matrix. The defaults to <code>NULL</code> , which only uses the systematic parameters (for which <code>DepVar = TRUE</code> in the appropriate <code>describe.model</code> function)
<code>...</code>	additional arguments passed to <code>model.matrix.default</code>

Value

A design matrix or array, depending on the options chosen in `shape`, with appropriate terms attributes.

Author(s)

Kosuke Imai <(kimai@princeton.edu)>; Gary King <(king@harvard.edu)>; Olivia Lau <(olau@fas.harvard.edu)>; Ferdinand Alimadhi <(falimadhi@iq.harvard.edu)>

See Also

`parse.par`, `parse.formula` and the full Zelig manual at <http://gking.harvard.edu/zelig>

Examples

```
# Let's say that the name of the model is "bivariate.probit", and
# the corresponding describe function is describe.bivariate.probit(),
# which identifies mu1 and mu2 as systematic components, and an
# ancillary parameter rho, which may be parameterized, but is estimated
# as a scalar by default. Let par be the parameter vector (including
# parameters for rho), formulae a user-specified formula, and mydata
# the user specified data frame.

# Acceptable combinations of parse.par() and model.matrix() are as follows:
## Setting up
## Not run:
data(sanction)
formulae <- cbind(import, export) ~ coop + cost + target
fml <- parse.formula(formulae, model = "bivariate.probit")
D <- model.frame(fml, data = sanction)
terms <- attr(D, "terms")

## Intuitive option
Beta <- parse.par(par, terms, shape = "vector", eqn = c("mu1", "mu2"))
X <- model.matrix(fml, data = D, shape = "stacked", eqn = c("mu1", "mu2"))
eta <- X

## Memory-efficient (compact) option (default)
Beta <- parse.par(par, terms, eqn = c("mu1", "mu2"))
X <- model.matrix(fml, data = D, eqn = c("mu1", "mu2"))
eta <- X

## Computationally-efficient (array) option
Beta <- parse.par(par, terms, shape = "vector", eqn = c("mu1", "mu2"))
X <- model.matrix(fml, data = D, shape = "array", eqn = c("mu1", "mu2"))
eta <- apply(X, 3, '
## End(Not run)
```

Description

This function accepts individual matrices as its inputs, combining the input matrices into a single data frame which can then be used in the `data` argument for social network analysis (models "netlm" and "netlogit") in Zelig.

Usage

```
network(...)
```

Arguments

... matrices representing variables, with rows and columns corresponding to individuals. These can be given as named arguments and should be given in the order the in which the user wishes them to appear in the output data frame.

Value

The `network` function creates a data frame which contains matrices instead of vectors as its variables. Inputs to the function should all be square matrices and can be given as named arguments.

Author(s)

Skyler J. Cranmer

See Also

The full Zelig manual is available at <http://gking.harvard.edu/zelig>.

Examples

```
## Not run:  
## Let Var1, Var2, Var3, Var4, and Var5 be matrices  
friendship <- network(Var1, Var2, Var3, Var4, Var5)  
## End(Not run)
```

newpainters

The Discretized Painter's Data of de Piles

Description

The original painters data contain the subjective assessment, on a 0 to 20 integer scale, of 54 classical painters. The `newpainters` data discretizes the subjective assessment by quartiles with thresholds 25%, 50%, 75%. The painters were assessed on four characteristics: composition, drawing, colour and expression. The data is due to the Eighteenth century art critic, de Piles.

Usage

```
data(newpainters)
```

Format

A table containing 5 variables ("Composition", "Drawing", "Colour", "Expression", and "School") and 54 observations.

Source

A. J. Weekes (1986). "A Genstat Primer". Edward Arnold.

M. Davenport and G. Studdert-Kennedy (1972). "The statistical analysis of aesthetic judgement: an exploration." *Applied Statistics*, vol. 21, pp. 324–333.

I. T. Jolliffe (1986) "Principal Component Analysis." Springer.

References

Venables, W. N. and Ripley, B. D. (2002) "Modern Applied Statistics with S," Fourth edition. Springer.

parse.formula	<i>Parsing user-input formulas into multiple syntax</i>
---------------	---

Description

Parse the input formula (or list of formulas) into the standard format described below. Since labels for this format will vary by model, `parse.formula` will evaluate a function `describe.model`, where `model` is given as an input to `parse.formula`.

If the `describe.model` function has more than one parameter for which `ExpVar = TRUE` and `DepVar = TRUE`, then the user-specified equations must have labels to match those parameters, else `parse.formula` should return an error. In addition, if the formula entries are not unambiguous, then `parse.formula` returns an error.

Usage

```
parse.formula(formula, model, data = NULL)
```

Arguments

<code>formula</code>	either a single formula or a list of <code>formula</code> objects
<code>model</code>	a character string specifying the name of the model
<code>data</code>	an optional data frame for models that require a factor response variable

Details

Acceptable user inputs are as follows:

	User Input	Output from <code>parse.formula</code>
Same covariates, separate effects	<code>cbind(y1, y2) ~ x1 + x2 * x3</code>	<code>list(mu1 = y1 ~ x1 + x2 * x3, mu2 = y2 ~ x1 + x2 * x3, rho = ~ 1)</code>
With <code>rho</code> as a systematic equation	<code>list(cbind(y1, y2) ~ x1 + x2, rho = ~ x4 + x5)</code>	<code>list(mu1 = y1 ~ x1 + x2, mu2 = y2 ~ x1 + x2, rho = ~ x4 + x5)</code>
With constraints (same variable)	<code>list(mu1 = y1 ~ x1 + tag(x2, "x2"), mu2 = y2 ~ x3 + tag(x2, "x2"))</code>	<code>list(mu1 = y1 ~ x1 + tag(x2, "x2"), mu2 = y2 ~ x3 + tag(x2, "x2"), rho = ~ 1)</code>
With constraints (different variables)	<code>list(mu1 = y1 ~ x1 + tag(x2, "z1"), mu2 = y2 ~ x3 + tag(x4, "z1"))</code>	<code>list(mu1 = y1 ~ x1 + tag(x2, "z1"), mu2 = y2 ~ x3 + tag(x4, "z1"), rho = ~ 1)</code>

Value

The output is a list of formula objects with class `c("multiple", "list")`. Let's say that the name of the model is `"bivariate.probit"`, and the corresponding describe function is `describe.bivariate.probit`, which identifies `mu1` and `mu2` as systematic components, and an ancillary parameter `rho`, which may be parameterized, but is estimated as a scalar by default.

Author(s)

Kosuke Imai <(kimai@princeton.edu)>; Gary King <(king@harvard.edu)>; Olivia Lau <(olau@fas.harvard.edu)>; Ferdinand Alimadhi <(falimadhi@iq.harvard.edu)>

See Also

`parse.par`, `model.frame.multiple`, `model.matrix.multiple`, and the full Zelig manual at <http://gking.harvard.edu/zelig>.

Examples

```
## Not run:
data(sanction)
formulae <- list(cbind(import, export) ~ coop + cost + target)
fml <- parse.formula(formulae, model = "bivariate.probit")
D <- model.frame(fml, data = sanction)
## End(Not run)
```

`parse.par`*Select and reshape parameter vectors*

Description

The `parse.par` function reshapes parameter vectors for comfortability with the output matrix from `model.matrix.multiple`. Use `parse.par` to identify sets of parameters; for example, within optimization functions that require vector input, or within `qi` functions that take matrix input of all parameters as a lump.

Usage

```
parse.par(par, terms, shape = "matrix", eqn = NULL)
```

Arguments

<code>par</code>	the vector (or matrix) of parameters
<code>terms</code>	the terms from either <code>model.frame.multiple</code> or <code>model.matrix.multiple</code>
<code>shape</code>	a character string (either "matrix" or "vector") that identifies the type of output structure
<code>eqn</code>	a character string (or strings) that identify the parameters that you would like to subset from the larger <code>par</code> structure

Value

A matrix or vector of the sub-setted (and reshaped) parameters for the specified parameters given in "eqn". By default, `eqn = NULL`, such that all systematic components are selected. (Systematic components have `ExpVar = TRUE` in the appropriate `describe.model` function.)

If an ancillary parameter (for which `ExpVar = FALSE` in `describe.model`) is specified in `eqn`, it is always returned as a vector (ignoring `shape`). (Ancillary parameters are all parameters that have intercept only formulas.)

Author(s)

Kosuke Imai <kimai@princeton.edu>; Gary King <king@harvard.edu>; Olivia Lau <olau@fas.harvard.edu>; Ferdinand Alimadhi <falimadhi@iq.harvard.edu>

See Also

`model.matrix.multiple`, `parse.formula` and the full Zelig manual at <http://gking.harvard.edu/zelig>

Examples

```

# Let's say that the name of the model is "bivariate.probit", and
# the corresponding describe function is describe.bivariate.probit(),
# which identifies mu1 and mu2 as systematic components, and an
# ancillary parameter rho, which may be parameterized, but is estimated
# as a scalar by default. Let par be the parameter vector (including
# parameters for rho), formulae a user-specified formula, and mydata
# the user specified data frame.

# Acceptable combinations of parse.par() and model.matrix() are as follows:
## Setting up
## Not run:
data(sanction)
formulae <- cbind(import, export) ~ coop + cost + target
fml <- parse.formula(formulae, model = "bivariate.probit")
D <- model.frame(fml, data = sanction)
terms <- attr(D, "terms")

## Intuitive option
Beta <- parse.par(par, terms, shape = "vector", eqn = c("mu1", "mu2"))
X <- model.matrix(fml, data = D, shape = "stacked", eqn = c("mu1", "mu2"))
eta <- X

## Memory-efficient (compact) option (default)
Beta <- parse.par(par, terms, eqn = c("mu1", "mu2"))
X <- model.matrix(fml, data = D, eqn = c("mu1", "mu2"))
eta <- X

## Computationally-efficient (array) option
Beta <- parse.par(par, terms, shape = "vector", eqn = c("mu1", "mu2"))
X <- model.matrix(fml, data = D, shape = "array", eqn = c("mu1", "mu2"))
eta <- apply(X, 3, '
## End(Not run)

```

 PERisk

Political Economic Risk Data from 62 Countries in 1987

Description

Political Economic Risk Data from 62 Countries in 1987.

Usage

```
data(PERisk)
```

Format

A data frame with 62 observations on the following 6 variables. All data points are from 1987. See Quinn (2004) for more details.

country: a factor with levels 'Argentina' 'Australia' 'Austria' 'Bangladesh' 'Belgium' 'Bolivia' 'Botswana' 'Brazil' 'Burma' 'Cameroon' 'Canada' 'Chile' 'Colombia' 'Congo-Kinshasa' 'Costa Rica' 'Cote d'Ivoire' 'Denmark' 'Dominican Republic' 'Ecuador' 'Finland' 'Gambia, The' 'Ghana' 'Greece' 'Hungary' 'India' 'Indonesia' 'Iran' 'Ireland' 'Israel' 'Italy' 'Japan' 'Kenya' 'Korea, South' 'Malawi' 'Malaysia' 'Mexico' 'Morocco' 'New Zealand' 'Nigeria' 'Norway' 'Papua New Guinea' 'Paraguay' 'Philippines' 'Poland' 'Portugal' 'Sierra Leone' 'Singapore' 'South Africa' 'Spain' 'Sri Lanka' 'Sweden' 'Switzerland' 'Syria' 'Thailand' 'Togo' 'Tunisia' 'Turkey' 'United Kingdom' 'Uruguay' 'Venezuela' 'Zambia' 'Zimbabwe'

courts: an ordered factor with levels '0' < '1'. 'courts' is an indicator of whether the country in question is judged to have an independent judiciary. From Henisz (2002).

barb2: a numeric vector giving the natural log of the black market premium in each country. The black market premium is coded as the black market exchange rate (local currency per dollar) divided by the official exchange rate minus 1. From Marshall, Gurr, and Harff (2002).

prsexp2: an ordered factor with levels '0' < '1' < '2' < '3' < '4' < '5', giving the lack of expropriation risk. From Marshall, Gurr, and Harff (2002).

prscorr2: an ordered factor with levels '0' < '1' < '2' < '3' < '4' < '5', measuring the lack of corruption. From Marshall, Gurr, and Harff (2002).

gdpw2: a numeric vector giving the natural log of real GDP per worker in 1985 international prices. From Alvarez et al. (1999).

Source

Mike Alvarez, Jose Antonio Cheibub, Fernando Limongi, and Adam Przeworski. 1999. "ACLP Political and Economic Database." <URL: <http://www.ssc.upenn.edu/~cheibub/data/>>.

Witold J. Henisz. 2002. "The Political Constraint Index (POLCON) Dataset." <URL: <http://www-management.wharton.upenn.edu/henisz/POLCON/ContactInfo.html>>.

Monty G. Marshall, Ted Robert Gurr, and Barbara Harff. 2002. "State Failure Task Force Problem Set." <URL: <http://www.cidcm.umd.edu/inscr/stfail/index.htm>>.

References

Kevin M. Quinn. 2004. "Bayesian Factor Analysis for Mixed Ordinal and Continuous Response." *Political Analysis*. Vol. 12, pp.338–353.

plot.ci

Plotting Vertical confidence Intervals

Description

The `plot.ci` command generates vertical confidence intervals for linear or generalized linear univariate response models.

Usage

```
plot.ci(x, CI = 95, qi = "ev", main = "", ylab = NULL, xlab = NULL,
        xlim = NULL, ylim = NULL, col = c("red", "blue"), ...)
```

Arguments

<code>x</code>	stored output from <code>sim</code> . The <code>x\$x</code> and optional <code>x\$x1</code> values used to generate the <code>sim</code> output object must have more than one observation.
<code>CI</code>	the selected confidence interval. Defaults to 95 percent.
<code>qi</code>	the selected quantity of interest. Defaults to expected values.
<code>main</code>	a title for the plot.
<code>ylab</code>	label for the y-axis.
<code>xlab</code>	label for the x-axis.
<code>xlim</code>	limits on the x-axis.
<code>ylim</code>	limits on the y-axis.
<code>col</code>	a vector of at most two colors for plotting the expected value given by <code>x</code> and the alternative set of expected values given by <code>x1</code> in <code>sim</code> . If the quantity of interest selected is not the expected value, or <code>x1 = NULL</code> , only the first color will be used.
<code>...</code>	Additional parameters passed to <code>plot</code> .

Value

For all univariate response models, `plot.ci()` returns vertical confidence intervals over a specified range of one explanatory variable. You may save this plot using the commands described in the Zelig manual (<http://gking.harvard.edu/zelig>).

Author(s)

Kosuke Imai <kimai@princeton.edu>; Gary King <king@harvard.edu>; Olivia Lau <olau@fas.harvard.edu>

See Also

The full Zelig manual is available at <http://gking.harvard.edu/zelig>, and users may also wish to see `plot`, `lines`.

Examples

```
data(turnout)
z.out <- zelig(vote ~ race + educate + age + I(age^2) + income,
              model = "logit", data = turnout)
age.range <- 18:95
x.low <- setx(z.out, educate = 12, age = age.range)
x.high <- setx(z.out, educate = 16, age = age.range)
s.out <- sim(z.out, x = x.low, x1 = x.high)
plot.ci(s.out, xlab = "Age in Years",
        ylab = "Predicted Probability of Voting",
        main = "Effect of Education and Age on Voting Behavior")
legend(45, 0.52, legend = c("College Education (16 years)",
                           "High School Education (12 years)"), col = c("blue", "red"),
      lty = c("solid"))
```

plot.surv

Plotting Confidence Intervals for Survival Curves

Description

The `plot.surv` command generates confidence intervals for Kaplan-Meier survival curves

Usage

```
plot.surv(x, duration, censor, type = "line", plotcensor=TRUE,
          plottimes = FALSE, int = c(0.025,0.975), ...)
```

Arguments

<code>x</code>	output from <code>sim</code> stored as a list. Each element of the list is the <code>sim</code> output for a particular survival curve.
<code>duration</code>	the duration variable (e.g. lifetime, survival, etc.).
<code>censor</code>	the censored data
<code>type</code>	the type of confidence interval. Defaults to "line", which draws vertical confidence intervals at observed event times. "poly" draws confidence regions using polygons.
<code>plotcensor</code>	default is TRUE. Plots censoring times as a rug object.
<code>plottimes</code>	default is FALSE. Plots step function with indicators at observed event times.
<code>int</code>	vector of quantile limits for the confidence interval. Default is 95% interval.
<code>...</code>	Additional parameters passed to <code>plot</code> .

Value

For survival models, `plot.surv()` returns vertical confidence intervals or polygon survival regions for Kaplan-Meier survival curves. You may save this plot using the commands described in the Zelig manual (<http://gking.harvard.edu/zelig>).

Author(s)

John A. Graves <graveja0@gmail.com>

See Also

The full Zelig manual is available at <http://gking.harvard.edu/zelig>, and users may also wish to see `plot`, `lines`.

Examples

```
## Not run:
data(coalition)
z.out1 <- zelig(Surv(duration, ciepl2)~invest+numst2+crisis,
robust=TRUE, cluster="polar", model="coxph", data=coalition)
low <- setx(z.out1, numst2=0)
high <- setx(z.out1, numst2=1)
# Simulate Survival Curves for Each Group
s.out1 <- sim(z.out1, x=low)
s.out2 <- sim(z.out1, x=high)

# Organize simulated output as a list
out <- list(s.out1, s.out2)

plot.surv(x = out, duration = coalition$duration, censor=coalition$ciepl2,
type="line", plottimes=FALSE, plotcensor=FALSE,
main="Survival", xlab="Time", ylab="Survival")
## End(Not run)
```

plot.zelig

*Graphing Quantities of Interest***Description**

The `zelig` method for the generic `plot` command generates default plots for `sim` output with one-observation values in `x` and `x1`.

Usage

```
## S3 method for class 'zelig':
plot(x, xlab = "", user.par = FALSE, ...)
```

Arguments

<code>x</code>	stored output from <code>sim</code> . If the <code>x\$x</code> or <code>x\$x1</code> values stored in the object contain more than one observation, <code>plot.zelig</code> will return an error. For linear or generalized linear models with more than one observation in <code>x\$x</code> and optionally <code>x\$x1</code> , you may use <code>plot.ci</code> .
<code>xlab</code>	a character string for the x-axis label for all graphs.
<code>user.par</code>	a logical value indicating whether to use the default Zelig plotting parameters (<code>user.par = FALSE</code>) or user-defined parameters (<code>user.par = TRUE</code>), set using the <code>par</code> function prior to plotting.
<code>...</code>	Additional parameters passed to <code>plot.default</code> . Because <code>plot.zelig</code> primarily produces diagnostic plots, many of these parameters are hard-coded for convenience and presentation.

Value

Depending on the class of model selected, `plot.zelig` will return an on-screen window with graphs of the various quantities of interest. You may save these plots using the commands described in the Zelig manual (available at <http://gking.harvard.edu/zelig>).

Author(s)

Kosuke Imai <(kimai@princeton.edu)>; Gary King <(king@harvard.edu)>; Olivia Lau <(olau@fas.harvard.edu)>

See Also

The full Zelig manual at <http://gking.harvard.edu/zelig> and `plot`, `lines`, and `par`.

`put.start`

Set specific starting values for certain parameters

Description

After calling `set.start` to create default starting values, use `put.start` to change starting values for specific parameters or parameter sets.

Usage

```
put.start(start.val, value, terms, eqn)
```

Arguments

<code>start.val</code>	the vector of starting values created by <code>set.start</code>
<code>value</code>	the scalar or vector of replacement starting values
<code>terms</code>	the terms output from <code>model.frame.multiple</code>
<code>eqn</code>	character vector of the parameters for which you would like to replace the default values with <code>value</code>

Value

A vector of starting values (of the same length as `start.val`)

Author(s)

Kosuke Imai <(kimai@princeton.edu)>; Gary King <(king@harvard.edu)>; Olivia Lau <(olau@fas.harvard.edu)>; Ferdinand Alimadhi <(falimadhi@iq.harvard.edu)>

See Also

`set.start`, and the full Zelig manual at <http://gking.harvard.edu/zelig>.

 repl *Replicating Analyses*

Description

The generic function `repl` command takes `zelig` or `sim` output objects and replicates (literally, re-runs) the entire analysis. The results should be an output object identical to the original input object in the case of `zelig` output. In the case of `sim` output, the replicated analyses may differ slightly due to stochastic randomness in the simulation procedure.

Usage

```
repl(object, data, ...)
## Default S3 method:
repl(object, data = NULL, ...)
## S3 method for class 'zelig':
repl(object, data = NULL, prev = NULL, x = NULL, x1 = NULL,
      bootfn = NULL, ...)
```

Arguments

<code>object</code>	Stored output from either <code>zelig</code> or <code>sim</code> .
<code>data</code>	You may manually input the data frame name rather than allowing <code>repl</code> to draw the data frame name from the object to be replicated.
<code>prev</code>	When replicating <code>sim</code> output, you may optionally use the previously simulated parameters to calculate the quantities of interest rather than simulating a new set of parameters. For all models, this should produce identical quantities of interest. In addition, for if the parameters were bootstrapped in the original analysis, this will save a considerable amount of time.
<code>x</code>	When replicating <code>sim</code> output, you may optionally use an alternative <code>setx</code> value for the <code>x</code> input.
<code>x1</code>	When replicating <code>sim</code> output, you may optionally use an alternative <code>setx</code> object for the <code>x1</code> input to replicating the <code>sim</code> object.
<code>bootfn</code>	When replicating <code>sim</code> output with bootstrapped parameters, you should manually specify the <code>bootfn</code> if a non-default option was used.
<code>...</code>	Additional arguments passed to either <code>zelig</code> or <code>sim</code> .

Value

For `zelig` output, `repl` will create output that is in every way identical to the original input. You may check to see whether they are identical by using the `identical` command.

For `sim` output, `repl` output will be identical to the original object if you choose not to simulate new parameters, and instead choose to calculate quantities of interest using the previously simulated parameters (using the `prev` option. If you choose to simulate new parameters, the summary statistics for each quantity of interest should be identical, up to a random approximation error. As the number of simulations increases, this error decreases.

Author(s)

Kosuke Imai <(kimai@princeton.edu)>; Gary King <(king@harvard.edu)>; Olivia Lau <(olau@fas.harvard.edu)>

See Also

`zelig`, `setx`, and `sim`. In addition, the full Zelig manual may be accessed online at <http://gking.harvard.edu/zelig>.

Examples

```
data(turnout)
z.out <- zelig(vote ~ race + educate, model = "logit", data = turnout[1:1000,])
x.out <- setx(z.out)
s.out <- sim(z.out, x = x.out)
z.rep <- repl(z.out)
identical(z.out$coef, z.rep$coef)
z.alt <- repl(z.out, data = turnout[1001:2000,])
s.rep <- repl(s.out, prev = s.out$par)
identical(s.out$ev, s.rep$ev)
```

 rocplot

Receiver Operator Characteristic Plots

Description

The `rocplot` command generates a receiver operator characteristic plot to compare the in-sample (default) or out-of-sample fit for two logit or probit regressions.

Usage

```
rocplot(y1, y2, fitted1, fitted2, cutoff = seq(from=0, to=1, length=100),
        lty1 = "solid", lty2 = "dashed", lwd1 = par("lwd"), lwd2 = par("lwd"),
        col1 = par("col"), col2 = par("col"), main, xlab, ylab,
        plot = TRUE, ...)
```

Arguments

<code>y1</code>	Response variable for the first model.
<code>y2</code>	Response variable for the second model.
<code>fitted1</code>	Fitted values for the first model. These values may represent either the in-sample or out-of-sample fitted values.
<code>fitted2</code>	Fitted values for the second model.
<code>cutoff</code>	A vector of cut-off values between 0 and 1, at which to evaluate the proportion of 0s and 1s correctly predicted by the first and second model. By default, this is 100 increments between 0 and 1, inclusive.
<code>lty1, lty2</code>	The line type for the first model (<code>lty1</code>) and the second model (<code>lty2</code>), defaulting to solid and dashed, respectively.

<code>lwd1, lwd2</code>	The width of the line for the first model (<code>lwd1</code>) and the second model (<code>lwd2</code>), defaulting to 1 for both.
<code>col1, col2</code>	The colors of the line for the first model (<code>col1</code>) and the second model (<code>col2</code>), defaulting to black for both.
<code>main</code>	a title for the plot. Defaults to <code>ROC Curve</code> .
<code>xlab</code>	a label for the x-axis. Defaults to <code>Proportion of 1's Correctly Predicted</code> .
<code>ylab</code>	a label for the y-axis. Defaults to <code>Proportion of 0's Correctly Predicted</code> .
<code>plot</code>	defaults to <code>TRUE</code> , which generates a plot to the selected device. If <code>FALSE</code> , returns a list of items (see below).
<code>...</code>	Additional parameters passed to <code>plot</code> , including <code>xlab</code> , <code>ylab</code> , and <code>main</code> .

Value

If `plot = TRUE`, `rocplot` generates an ROC plot for two logit or probit models. If `plot = FALSE`, `rocplot` returns a list with the following elements:

<code>roc1</code>	a matrix containing a vector of x-coordinates and y-coordinates corresponding to the number of ones and zeros correctly predicted for the first model.
<code>roc2</code>	a matrix containing a vector of x-coordinates and y-coordinates corresponding to the number of ones and zeros correctly predicted for the second model.
<code>area1</code>	the area under the first ROC curve, calculated using Reimann sums.
<code>area2</code>	the area under the second ROC curve, calculated using Reimann sums.

Author(s)

Kosuke Imai <kimai@princeton.edu>; Gary King <king@harvard.edu>; Olivia Lau <olau@fas.harvard.edu>

See Also

The full Zelig manual (available at <http://gking.harvard.edu/zelig>), `plot`, `lines`.

Examples

```
data(turnout)
z.out1 <- zelig(vote ~ race + educate + age, model = "logit",
  data = turnout)
z.out2 <- zelig(vote ~ race + educate, model = "logit",
  data = turnout)
rocplot(z.out1$y, z.out2$y, fitted(z.out1), fitted(z.out2))
```

sanction	<i>Multilateral Economic Sanctions</i>
----------	--

Description

Data on bilateral sanctions behavior for selected years during the general period 1939-1983. This data contains errors that have since been corrected. Please contact Lisa Martin before using this data for publication.

Usage

```
data(sanction)
```

Format

A table containing 8 variables ("mil", "coop", "target", "import", "export", "cost", "num", and "ncost") and 78 observations. For full variable description, see Martin, 1992.

Source

Martin, 1992

References

Martin, Lisa (1992). *Coercive Cooperation: Explaining Multilateral Economic Sanctions*, Princeton: Princeton University Press.

set.start	<i>Set starting values for all parameters</i>
-----------	---

Description

After using `parse.par` and `model.matrix.multiple`, use `set.start` to set starting values for all parameters. By default, starting values are set to 0. If you wish to select alternative starting values for certain parameters, use `put.start` after `set.start`.

Usage

```
set.start(start.val = NULL, terms)
```

Arguments

<code>start.val</code>	user-specified starting values. If <code>NULL</code> (default), the default starting values for all parameters are set to 0.
<code>terms</code>	the terms output from <code>model.frame.multiple</code>

Value

A named vector of starting values for all parameters specified in `terms`, defaulting to 0.

Author(s)

Kosuke Imai <(kimai@princeton.edu)>; Gary King <(king@harvard.edu)>; Olivia Lau <(olau@fas.harvard.edu)>; Ferdinand Alimadhi <(falimadhi@iq.harvard.edu)>

See Also

`put.start`, `parse.par`, `model.frame.multiple`, and the full Zelig manual at <http://gking.harvard.edu/zelig>.

Examples

```
## Not run:
fml <- parse.formula(formula, model = "bivariate.probit")
D <- model.frame(fml, data = data)
terms <- attr(D, "terms")
start.val <- set.start(start.val = NULL, terms)
## End(Not run)
```

 setx

Setting Explanatory Variable Values

Description

The `setx` command uses the variables identified in the `formula` generated by `zelig` and sets the values of the explanatory variables to the selected values. Use `setx` after `zelig` and before `sim` to simulate quantities of interest.

Usage

```
x.out <- setx(object, fn = list(numeric = mean, ordered = median,
                              others = mode),
             data = NULL, cond = FALSE, ...)
```

Arguments

<code>object</code>	the saved output from <code>zelig</code> .
<code>fn</code>	a list of functions to apply to three types of variables:
<code>numeric</code>	numeric variables are set to their mean by default, but you may select any mathematical function to apply to numeric variables.
<code>ordered</code>	ordered factors are set to their median by default, and most mathematical operations will work on them. If you select <code>ordered = mean</code> , however, <code>setx</code> will default to median with a warning.

other variables may consist of unordered factors, character strings, or logical variables. The other variables may only be set to their mode. If you wish to set one of the other variables to a specific value, you may do so using . . . below.

In the special case `fn = NULL`, `setx` will return all of the observations without applying any function to the data.

<code>data</code>	a new data frame used to set the values of explanatory variables. If <code>data = NULL</code> (the default), the data frame called in <code>zelig</code> is used.
<code>cond</code>	a logical value indicating whether unconditional (default) or conditional (choose <code>cond = TRUE</code>) prediction should be performed. If you choose <code>cond = TRUE</code> , <code>setx</code> will coerce <code>fn = NULL</code> and ignore the additional arguments in If <code>cond = TRUE</code> and <code>data = NULL</code> , <code>setx</code> will prompt you for a data frame.
<code>. . .</code>	user-defined values of specific variables overwriting the default values set by the function <code>fn</code> . For example, adding <code>var1 = mean(data\$var1)</code> or <code>x1 = 12</code> explicitly sets the value of <code>x1</code> to 12. In addition, you may specify one explanatory variable as a range of values, creating one observation for every unique value in the range of values.

Value

For unconditional prediction, `x.out` is a model matrix based on the specified values for the explanatory variables. For multiple analyses (i.e., when choosing the `by` option in `zelig`, `setx` returns the selected values calculated over the entire data frame. If you wish to calculate values over just one subset of the data frame, the 5th subset for example, you may use: `x.out <- setx(z.out[[5]])`

For conditional prediction, `x.out` includes the model matrix and the dependent variables. For multiple analyses (when choosing the `by` option in `zelig`), `setx` returns the observed explanatory variables in each subset.

Author(s)

Kosuke Imai <(kimai@princeton.edu)>; Gary King <(king@harvard.edu)>; Olivia Lau <(olau@fas.harvard.edu)>

See Also

The full Zelig manual may be accessed online at <http://gking.harvard.edu/zelig>.

Examples

```
# Unconditional prediction:
data(turnout)
z.out <- zelig(vote ~ race + educate, model = "logit", data = turnout)
x.out <- setx(z.out)
s.out <- sim(z.out, x = x.out)

# Unconditional prediction with all observations:
x.out <- setx(z.out, fn = NULL)
s.out <- sim(z.out, x = x.out)
```

```

# Unconditional prediction with out of sample data:
z.out <- zelig(vote ~ race + educate, model = "logit",
              data = turnout[1:1000,])
x.out <- setx(z.out, data = turnout[1001:2000,])
s.out <- sim(z.out, x = x.out)

# Using a user-defined function in fn:
## Not run:
quants <- function(x)
  quantile(x, 0.25)
x.out <- setx(z.out, fn = list(numeric = quants))
## End(Not run)

# Conditional prediction:
## Not run:
library(MatchIt)
data(lalonde)
match.out <- matchit(treat ~ age + educ + black + hispan + married +
                    nodegree + re74 + re75, data = lalonde)
z.out <- zelig(re78 ~ distance, data = match.data(match.out, "control"),
              model = "ls")
x.out <- setx(z.out, fn = NULL, data = match.data(match.out, "treat"),
              cond = TRUE)
s.out <- sim(z.out, x = x.out)
## End(Not run)

```

sim

Simulating Quantities of Interest

Description

Simulate quantities of interest from the estimated model output from `zelig()` given specified values of explanatory variables established in `setx()`. For classical *maximum likelihood* models, `sim()` uses asymptotic normal approximation to the log-likelihood. For *Bayesian models*, Zelig simulates quantities of interest from the posterior density, whenever possible. For *robust Bayesian models*, simulations are drawn from the identified class of Bayesian posteriors. Alternatively, you may generate quantities of interest using bootstrapped parameters.

Usage

```
s.out <- sim(object, x, x1 = NULL, num = c(1000, 100), prev = NULL,
            bootstrap = FALSE, bootfn = NULL, ...)
```

Arguments

<code>object</code>	the output object from <code>zelig</code> .
<code>x</code>	values of explanatory variables used for simulation, generated by <code>setx</code> .

x1	optional values of explanatory variables (generated by a second call of <code>setx</code>), used to simulate first differences and risk ratios. (Not available for conditional prediction.)
num	the number of simulations, i.e., posterior draws. If the <code>num</code> argument is omitted, <code>sim</code> draws 1,000 simulations by if <code>bootstrap = FALSE</code> (the default), or 100 simulations if <code>bootstrap = TRUE</code> . You may increase this value to improve accuracy. (Not available for conditional prediction.)
bootstrap	a logical value indicating if parameters should be generated by re-fitting the model for bootstrapped data, rather than from the likelihood or posterior. (Not available for conditional prediction.)
bootfn	a function which governs how the data is sampled, re-fits the model, and returns the bootstrapped model parameters. If <code>bootstrap = TRUE</code> and <code>bootfn = NULL</code> , <code>sim</code> will sample observations from the original data (with replacement) until it creates a sampled dataset with the same number of observations as the original data. Alternative bootstrap methods include sampling the residuals rather than the observations, weighted sampling, and parametric bootstrapping. (Not available for conditional prediction.)
...	additional optional arguments passed to <code>boot</code> .

Value

The output stored in `s.out` varies by model. Use the `names` command to view the output stored in `s.out`. Common elements include:

x	the <code>setx</code> values for the explanatory variables, used to calculate the quantities of interest (expected values, predicted values, etc.).
x1	the optional <code>setx</code> object used to simulate first differences, and other model-specific quantities of interest, such as risk-ratios.
call	the options selected for <code>sim</code> , used to replicate quantities of interest.
zelig.call	the original command and options for <code>zelig</code> , used to replicate analyses.
num	the number of simulations requested.
par	the parameters (coefficients, and additional model-specific parameters). You may wish to use the same set of simulated parameters to calculate quantities of interest rather than simulating another set.
qi\$ev	simulations of the expected values given the model and <code>x</code> .
qi\$pr	simulations of the predicted values given by the fitted values.
qi\$fd	simulations of the first differences (or risk difference for binary models) for the given <code>x</code> and <code>x1</code> . The difference is calculated by subtracting the expected values given <code>x</code> from the expected values given <code>x1</code> . (If do not specify <code>x1</code> , you will not get first differences or risk ratios.)
qi\$rr	simulations of the risk ratios for binary and multinomial models. See specific models for details.
qi\$ate.ev	simulations of the average expected treatment effect for the treatment group, using conditional prediction. Let t_i be a binary explanatory variable defining

the treatment ($t_i = 1$) and control ($t_i = 0$) groups. Then the average expected treatment effect for the treatment group is

$$\frac{1}{n} \sum_{i=1}^n [Y_i(t_i = 1) - E[Y_i(t_i = 0) \mid t_i = 1]],$$

where $Y_i(t_i = 1)$ is the value of the dependent variable for observation i in the treatment group. Variation in the simulations are due to uncertainty in simulating $E[Y_i(t_i = 0)]$, the counterfactual expected value of Y_i for observations in the treatment group, under the assumption that everything stays the same except that the treatment indicator is switched to $t_i = 0$.

qi\$ate.pr

simulations of the average predicted treatment effect for the treatment group, using conditional prediction. Let t_i be a binary explanatory variable defining the treatment ($t_i = 1$) and control ($t_i = 0$) groups. Then the average predicted treatment effect for the treatment group is

$$\frac{1}{n} \sum_{i=1}^n [Y_i(t_i = 1) - Y_i(\widehat{t_i = 0}) \mid t_i = 1],$$

where $Y_i(t_i = 1)$ is the value of the dependent variable for observation i in the treatment group. Variation in the simulations are due to uncertainty in simulating $Y_i(\widehat{t_i = 0})$, the counterfactual predicted value of Y_i for observations in the treatment group, under the assumption that everything stays the same except that the treatment indicator is switched to $t_i = 0$.

In the case of censored Y in the exponential, Weibull, and lognormal models, `sim` first imputes the uncensored values for Y before calculating the ATE.

You may use the `$` operator to extract any of the above from `s.out`. For example, `s.outqiev` extracts the simulated expected values.

Author(s)

Kosuke Imai <(kimai@princeton.edu)>; Gary King <(king@harvard.edu)>; Olivia Lau <(olau@fas.harvard.edu)>

See Also

The full Zelig at <http://gking.harvard.edu/zelig>, and `boot`.

sna.ex

Simulated Example of Social Network Data

Description

This data set contains five sociomatrices of simulated data social network data.

Usage

`data(sna.ex)`

Format

Each variable in the dataset is a 25 by 25 matrix of simulated social network data. The matrices are labeled "Var1", "Var2", "Var3", "Var4", and "Var5".

Source

fictitious

summary.zelig *Summary of Simulated Quantities of Interest*

Description

Summarizes the object of class `zelig` (output from `sim`) which contains simulated quantities of interest.

Usage

```
## S3 method for class 'zelig':
summary(object, subset = NULL, CI = 95, stats = c("mean", "sd"), ...)
```

Arguments

<code>object</code>	output object from <code>sim</code> (of class "zelig").
<code>subset</code>	takes one of three values: <code>NULL</code> (default) for more than one observation, summarizes all the observations at once for each quantity of interest. a numeric vector indicates which observations to summarize, and summarizes each one independently. <code>all</code> summarizes all the observations independently.
<code>stats</code>	summary statistics to be calculated.
<code>CI</code>	a confidence interval to be calculated.
<code>...</code>	further arguments passed to or from other methods.

Value

<code>sim</code>	number of simulations, i.e., posterior draws.
<code>x</code>	values of explanatory variables used for simulation.
<code>x1</code>	values of explanatory variables used for simulation of first differences etc.
<code>qi.stats</code>	summary of quantities of interest. Use <code>names</code> to view the model-specific items available in <code>qi.stats</code> .

Author(s)

Kosuke Imai <(kimai@princeton.edu)>; Gary King <(king@harvard.edu)>; Olivia Lau <(olau@fas.harvard.edu)>

See Also

`zelig`, `setx`, `sim`, and `names`, and the full Zelig manual at <http://gking.harvard.edu/zelig>.

SupremeCourt

U.S. Supreme Court Vote Matrix

Description

This dataframe contains a matrix votes cast by U.S. Supreme Court justices in all cases in the 2000 term.

Usage

```
data(SupremeCourt)
```

Format

The dataframe has contains data for justices Rehnquist, Stevens, O'Connor, Scalia, Kennedy, Souter, Thomas, Ginsburg, and Breyer for the 2000 term of the U.S. Supreme Court. It contains data from 43 non-unanimous cases. The votes are coded liberal (1) and conservative (0) using the protocol of Spaeth (2003). The unit of analysis is the case citation (ANALU=0). We are concerned with formally decided cases issued with written opinions, after full oral argument and cases decided by an equally divided vote (DECTYPE=1,5,6,7).

Source

Harold J. Spaeth (2005). "Original United States Supreme Court Database: 1953-2004 Terms." <URL:<http://www.as.uky.edu/polisci/ulmerproject/sctdata.htm>>.

swiss

Swiss Fertility and Socioeconomic Indicators (1888) Data

Description

Standardized fertility measure and socio-economic indicators for each of 47 French-speaking provinces of Switzerland at about 1888.

Usage

```
data(swiss)
```

Format

A data frame with 47 observations on 6 variables, each of which is in percent, i.e., in [0,100].

[,1] Fertility Ig, "common standardized fertility measure" [,2] Agriculture [,3] Examination nation
[,4] Education [,5] Catholic [,6] Infant.Mortality live births who live less than 1 year.

All variables but 'Fert' give proportions of the population.

Source

Project "16P5", pages 549-551 in

Mosteller, F. and Tukey, J. W. (1977) "Data Analysis and Regression: A Second Course in Statistics". Addison-Wesley, Reading Mass.

indicating their source as "Data used by permission of Franice van de Walle. Office of Population Research, Princeton University, 1976. Unpublished data assembled under NICHD contract number No 1-HD-O-2077."

References

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) "The New S Language". Wadsworth & Brooks/Cole.

ternaryplot

Ternary diagram

Description

Visualizes compositional, 3-dimensional data in an equilateral triangle (from the vcd library, Version 0.1-3.3, Date 2004-04-21), using plot graphics. Differs from implementation in vcd (0.9-7), which uses grid graphics.

Usage

```
ternaryplot(x, scale = 1, dimnames = NULL, dimnames.position = c("corner", "edge", "r",
  dimnames.color = "black", id = NULL, id.color = "black", coordinates =
  grid = TRUE, grid.color = "gray", labels = c("inside", "outside", "none",
  labels.color = "darkgray", border = "black", bg = "white", pch = 19, ce
  prop.size = FALSE, col = "red", main = "ternary plot", ...)
```

Arguments

`x` a matrix with three columns.
`scale` row sums scale to be used.
`dimnames` dimension labels (defaults to the column names of `x`).
`dimnames.position`, `dimnames.color` position and color of dimension labels.

<code>id</code>	optional labels to be plotted below the plot symbols. <code>coordinates</code> and <code>id</code> are mutual exclusive.
<code>id.color</code>	color of these labels.
<code>coordinates</code>	if TRUE, the coordinates of the points are plotted below them. <code>coordinates</code> and <code>id</code> are mutual exclusive.
<code>grid</code>	if TRUE, a grid is plotted. May optionally be a string indicating the line type (default: "dotted").
<code>grid.color</code>	grid color.
<code>labels, labels.color</code>	position and color of the grid labels.
<code>border</code>	color of the triangle border.
<code>bg</code>	triangle background.
<code>pch</code>	plotting character. Defaults to filled dots.
<code>cex</code>	a numerical value giving the amount by which plotting text and symbols should be scaled relative to the default. Ignored for the symbol size if <code>prop.size</code> is not FALSE.
<code>prop.size</code>	if TRUE, the symbol size is plotted proportional to the row sum of the three variables, i.e. represents the weight of the observation.
<code>col</code>	plotting color.
<code>main</code>	main title.
<code>...</code>	additional graphics parameters (see <code>par</code>)

Details

A points' coordinates are found by computing the gravity center of mass points using the data entries as weights. Thus, the coordinates of a point $P(a,b,c)$, $a + b + c = 1$, are: $P(b + c/2, c * \sqrt{3}/2)$.

Author(s)

David Meyer
<david.meyer@ci.tuwien.ac.at>

References

M. Friendly (2000), *Visualizing Categorical Data*. SAS Institute, Cary, NC.

See Also

[ternarypoints](#)

Examples

```

data(mexico)
if (require(VGAM)) {
  z.out <- zelig(as.factor(vote88) ~ pristr + othcok + othsocok,
                model = "mlogit", data = mexico)
  x.out <- setx(z.out)
  s.out <- sim(z.out, x = x.out)

  ternaryplot(s.out$qi$ev, pch = ".", col = "blue",
              main = "1988 Mexican Presidential Election")
}

```

ternarypoints

Adding Points to Ternary Diagrams

Description

Use `ternarypoints` to add points to a ternary diagram generated using the `ternaryplot` function in the `vcd` library. Use ternary diagrams to plot expected values for multinomial choice models with three categories in the dependent variable.

Usage

```
ternarypoints(object, pch = 19, col = "blue", ...)
```

Arguments

<code>object</code>	The input object must be a matrix with three columns.
<code>pch</code>	The selected type of point. By default, <code>pch = 19</code> , solid disks.
<code>col</code>	The color of the points. By default, <code>col = "blue"</code> .
<code>...</code>	Additional parameters passed to <code>points</code> .

Value

The `ternarypoints` command adds points to a previously existing ternary diagram. Use `ternaryplot` in the `vcd` library to generate the main ternary diagram.

Author(s)

Kosuke Imai <kimai@princeton.edu>; Gary King <king@harvard.edu>; Olivia Lau <olau@fas.harvard.edu>

See Also

The full Zelig manual at <http://gking.harvard.edu/zelig>, `points`, and `ternaryplot`.

`tobin`*Tobin's Tobit Data*

Description

Economists fit a parametric censored data model called the 'tobit'. These data are from Tobin's original paper.

Usage

```
data(tobin)
```

Format

A data frame with 20 observations on the following 3 variables.

durable: Durable goods purchase

age: Age in years

quant: Liquidity ratio (x 1000)

Source

J. Tobin, Estimation of relationships for limited dependent variables, *Econometrica*, v26, 24-36, 1958.

`turnout`*Turnout Data Set from the National Election Survey*

Description

This data set contains individual-level turnout data. It pools several American National Election Surveys conducted during the 1992 presidential election year. Only the first 2,000 observations (from a total of 15,837 observations) are included in the sample data.

Usage

```
data(turnout)
```

Format

A table containing 5 variables ("race", "age", "educate", "income", and "vote") and 2,000 observations.

Source

National Election Survey

References

King, Gary, Michael Tomz, Jason Wittenberg (2000). "Making the Most of Statistical Analyses: Improving Interpretation and Presentation," *American Journal of Political Science*, vol. 44, pp.341–355.

```
user.prompt          Pause in demo files
```

Description

Use `user.prompt` while writing demo files to force users to hit return before continuing.

Usage

```
user.prompt ()
```

Author(s)

Olivia Lau <olau@fas.harvard.edu>

See Also

```
readline
```

Examples

```
## Not run:  
user.prompt ()  
## End(Not run)
```

```
voteincome          Sample Turnout and Demographic Data from the 2000 Current Population Survey
```

Description

This data set contains turnout and demographic data from a sample of respondents to the 2000 Current Population Survey (CPS). The states represented are South Carolina and Arkansas. The data represent only a sample and results from this example should not be used in publication.

Usage

```
data(voteincome)
```

Format

A data frame containing 7 variables ("state", "year", "vote", "income", "education", "age", "female") and 1500 observations.

state a factor variable with levels equal to "AR" (Arkansas) and "SC" (South Carolina)

year an integer vector

vote an integer vector taking on values "1" (Voted) and "0" (Did Not Vote)

income an integer vector ranging from "4" (Less than \$5000) to "17" (Greater than \$75000) denoting family income. See the CPS codebook for more information on variable coding

education an integer vector ranging from "1" (Less than High School Education) to "4" (More than a College Education). See the CPS codebook for more information on variable coding

age an integer vector ranging from "18" to "85"

female an integer vector taking on values "1" (Female) and "0" (Male)

Source

Census Bureau Current Population Survey

References

<http://www.census.gov/cps>

Weimar

1932 Weimar election data

Description

This data set contains election results for 10 kreise (equivalent to precincts) from the 1932 Weimar (German) election.

Usage

`data(Weimar)`

Format

A table containing 11 variables and 10 observations. The variables are

Nazi Number of votes for the Nazi party

Government Number of votes for the Government

Communists Number of votes for the Communist party

FarRight Number of votes for far right parties

Other Number of votes for other parties, and non-voters

shareunemployed Proportion unemployed

shareblue Proportion working class
sharewhite Proportion white-collar workers
sharedomestic Proportion domestic servants
shareprotestants Proportion Protestant

Source

ICPSR

zelig

Estimating a Statistical Model

Description

The `zelig` command estimates a variety of statistical models. Use `zelig` output with `setx` and `sim` to compute quantities of interest, such as predicted probabilities, expected values, and first differences, along with the associated measures of uncertainty (standard errors and confidence intervals).

Usage

```
z.out <- zelig(formula, model, data, by, save.data, cite, ...)
```

Arguments

<code>formula</code>	a symbolic representation of the model to be estimated, in the form $y \sim x_1 + x_2$, where y is the dependent variable and x_1 and x_2 are the explanatory variables, and y , x_1 , and x_2 are contained in the same dataset. (You may include more than two explanatory variables, of course.) The $+$ symbol means “inclusion” not “addition.” You may also include interaction terms and main effects in the form $x_1 * x_2$ without computing them in prior steps; $I(x_1 * x_2)$ to include only the interaction term and exclude the main effects; and quadratic terms in the form $I(x_1^2)$.
<code>model</code>	the name of a statistical model, enclosed in <code>" "</code> . Type <code>help.zelig("models")</code> to see a list of currently supported models.
<code>data</code>	the name of a data frame containing the variables referenced in the formula, or a list of multiply imputed data frames each having the same variable names and row numbers (created by <code>mi</code>).
<code>save.data</code>	If is set to <code>"TRUE"</code> , the input dataframe will be saved as an attribute (<code>"zelig.data"</code>) of the <code>zelig</code> output object.
<code>cite</code>	If is set to <code>"TRUE"</code> (default), the model citation will be printed out when this function is called.

`by` a factor variable contained in `data`. `Zelig` will subset the data frame based on the levels in the `by` variable, and estimate a model for each subset. This a particularly powerful option which will allow you to save a considerable amount of effort. For example, to run the same model on all fifty states, you could type: `z.out <- zelig(y ~ x1 + x2, data = mydata, model = "ls", by = "state")` You may also use `by` to run models using `MatchIt` subclass.

`...` additional arguments passed to `zelig`, depending on the model to be estimated.

Value

Depending on the class of model selected, `zelig` will return an object with elements including coefficients, residuals, and formula which may be summarized using `summary(z.out)` or individually extracted using, for example, `z.out$coefficients`. See the specific models listed above for additional output values, or simply type names (`z.out`).

Author(s)

Kosuke Imai <kimai@princeton.edu>; Gary King <king@harvard.edu>; Olivia Lau <olau@fas.harvard.edu>

See Also

The full Zelig manual is available at <http://gking.harvard.edu/zelig>.

<code>zelig.url</code>	<i>Table of links for Zelig</i>
------------------------	---------------------------------

Description

Table of linds for `help.zelig` for the core Zelig package.

<code>zeligDepStatus</code>	<i>Zelig Dependencies Packages Client Status</i>
-----------------------------	--

Description

Compares Zelig-matrix of dependencies to the locally installed packages. Finds those packages that **Zelig** depends on and are not installed in local environment. Also finds those packages that are locally installed but with lower versions than those required in the dependencies matrix. The Zelig-matrix includes any of dependency fields in the ‘DESCRIPTION’ files, i.e. depends, imports and suggests, for any packages directly derived from **Zelig** and for any of the models that **Zelig** supports.

Usage

```
zeligDepStatus(lib.loc = NULL)
```

Arguments

`lib.loc` a character vector of directory names of R libraries, or NULL. The default value NULL corresponds to all libraries currently known. If the default is used, the loaded packages are searched before the libraries.

Value

Returns a matrix of packages that are either installed locally with lower versions, or packages not installed but listed in the Zelig-matrix of dependencies. The matrix rows correspond to the packages and the columns contain the following fields

Package	names of packages.
Version	versions locally installed
Zideal	versions required in Zelig-matrix of dependencies.

Note

If the R version in the local environment is different from the R version that **Zelig** depends on, then, it is reported with a message and no further action is taken. If the installed packages have versions higher than the corresponding values in the Zelig-matrix, it is reported with a message.

Author(s)

Ferdinand Alimadhi and Elena Villalon

References

King, Gary. Zelig: Everyones Statistical Software. <http://gking.harvard.edu/zelig>.

See Also

[installed.packages](#) [packageDescription](#) [zeligDepUpdate](#)

Examples

```
## find packages in all libraries currently installed
## Not run: zstatus <- zeligDepStatus()
## find packages only in lib.loc
## Not run: zstatus <- zeligDepStatus(lib.loc=~/.R/mylibrary")
```

zeligDepUpdate *Download Zelig Dependencies Packages*

Description

Compares the packages in Zelig-matrix of dependencies to the locally installed packages. Finds local packages that have lower versions than in the corresponding row of the Zelig-matrix. Also, finds packages that **Zelig** required but are not installed locally. Downloads packages that are locally installed with lower versions and those from the Zelig-matrix that are not installed. The download repository is taken from either the default *repos* argument or from the 'URL' column of the Zelig-matrix of dependencies.

Usage

```
zeligDepUpdate(destdir = NULL, installWithVers = FALSE, lib.loc = NULL,  
repos = "http://cran.r-project.org")
```

Arguments

<code>destdir</code>	directory to store the compress source-codes of packages that are downloaded from web repositories.
<code>installWithVers</code>	if TRUE, will invoke the install of the package such that it can be referenced by package version.
<code>lib.loc</code>	character vector describing the location of R library trees to search through (and update packages therein).
<code>repos</code>	character vector, the base URL(s) of the repositories to use, i.e. the URL of the CRAN master such as "http://cran.r-project.org", which is the default, or its Statlib mirror, "http://lib.stat.cmu.edu/R/CRAN". Can be NULL to install from local zip files.

Value

No return value.

Note

Installs first level dependencies packages of **Zelig** using R function `install.packages` with the variable `dependencies` set equal to TRUE. If the installed packages have versions higher than the corresponding entry in Zelig-matrix, they are reported with a message. If the R version in the local environment is different from the R version that **Zelig** depends on, then, it is reported with a message and no further action is taken.

Author(s)

Ferdinand Alimadhi and Elena Villalon

References

King, Gary. Zelig: Everyones Statistical Software. <http://gking.harvard.edu/zelig>.

See Also

`zeligDepStatus install.packages`

Examples

```
##checks all libraries curently know for packages
## Not run: zeligDepUpdate()
##finds packages only in lib.loc
## Not run: zeligDepUpdate(lib.loc=~/.R/mylibrary")
```

```
zeligDescribeModelXML
      Zelig interface functions
```

Description

Zelig interface functions. Used by VDC DSB to communicate with Zelig.

Usage

```
zeligDescribeModelXML(modelName, force=FALSE, schemaVersion="1.1")
zeligInstalledModels(inZeligOnly=TRUE, schemaVersion="1.1")
zeligListModels(inZeligOnly=TRUE)
zeligModelDependency(modelName, repos)
zeligGetSpecial(modelName)
```

Arguments

<code>modelName</code>	Name of model as returned by <code>zeligInstalledModels</code> or <code>zeligListModels</code> .
<code>inZeligOnly</code>	Flag, include only models in official Zelig distribution
<code>repos</code>	URL of default repository to use
<code>schemaVersion</code>	version of Zelig schema
<code>force</code>	generate a description even if no custom description supplied

Value

Use `zeligInstalledModels` and `zeligListModels` to determine what models are available in zelig for a particular schema level. Use `zmodel2string(zeligDescribeModel())` to generate an XML instance describing a model. Use `zeligModelDependencies` to generate a list of package dependencies for models. Use `zeligGetSpecial` to get the name special function, if any, to apply to the outcome variables. All functions return NULL if results are not available for that model.

Author(s)

Micah Altman (thedata-userslists.sourceforge.net) <http://thedata.org>

See Also

[zelig](#)

Examples

```
## Not run:
# show all available models
zeligListModels(inZeligOnly=FALSE)
# show installed models
zeligInstalledModels()
# show dependency for normal.bayes
zeligModelDependency("normal.bayes", "http://cran.r-project.org/")
# description of logit
cat(zeligDescribeModelXML("ologit"))
# special function for factor analysis
zeligGetSpecial("factor.mix")
## End(Not run)
```

Index

*Topic **IO**

zeligDescribeModelXML, 57

*Topic **aplot**

ternarypoints, 49

*Topic **datasets**

approval, 5
bivariate, 5
coalition, 6
coalition2, 6
eidat, 9
free1, 9
free2, 11
friendship, 12
grunfeld, 13
hoff, 15
homerun, 16
immigration, 17
klein, 17
kmenta, 18
macro, 18
MatchIt.url, 19
mexico, 20
mid, 21
newpainters, 26
PErisk, 30
sanction, 39
sna.ex, 44
SupremeCourt, 46
swiss, 46
tobin, 50
turnout, 50
voteincome, 51
Weimar, 52
Zelig.url, 54

*Topic **documentation**

help.zelig, 15
zeligDepStatus, 54
zeligDepUpdate, 56

*Topic **file**

current.packages, 7
dims, 8
gsource, 14
repl, 36
rocplot, 37
setx, 40
sim, 42
summary.zelig, 45
user.prompt, 51
zelig, 53

*Topic **hplot**

plot.ci, 31
plot.surv, 33
plot.zelig, 34
ternaryplot, 47

*Topic **methods**

match.data, 19
mi, 20
network, 25

*Topic **package**

Zelig-package, 2

*Topic **print**

zeligDescribeModelXML, 57

*Topic **utilities**

model.end, 22
model.frame.multiple, 23
model.matrix.multiple, 24
parse.formula, 27
parse.par, 29
put.start, 35
set.start, 39

approval, 5

bivariate, 5

coalition, 6

coalition2, 6

current.packages, 7

dims, 8

- eidat, 9
- free1, 9
- free2, 11
- friendship, 12
- grunfeld, 13
- gsource, 14
- help.zelig, 15
- hoff, 15
- homerun, 16
- imm1 (*immigration*), 17
- imm2 (*immigration*), 17
- imm3 (*immigration*), 17
- imm4 (*immigration*), 17
- imm5 (*immigration*), 17
- immigration, 17
- install.packages, 57
- installed.packages, 55
- klein, 17
- kmenta, 18
- macro, 18
- match.data, 19
- MatchIt.url, 19
- mexico, 20
- mi, 20
- mid, 21
- model.end, 22
- model.frame, 22
- model.frame.default, 23
- model.frame.multiple, 23, 24, 28, 29, 35, 39, 40
- model.matrix.default, 23
- model.matrix.multiple, 24, 28, 29, 39
- names, 45, 46
- network, 25
- newpainters, 26
- optim, 22
- packageDescription, 55
- parse.formula, 23–25, 27, 29
- parse.par, 23, 25, 28, 29, 39, 40
- PERisk, 30
- plot (*plot.zelig*), 34
- plot.ci, 31, 34
- plot.surv, 33
- plot.zelig, 34
- put.start, 35, 39, 40
- repl, 36
- ROC (*rocplot*), 37
- roc (*rocplot*), 37
- ROCplot (*rocplot*), 37
- rocplot, 37
- sanction, 39
- set.start, 35, 39
- setx, 36, 37, 40, 42, 43, 46
- sim, 34, 36, 37, 42, 43, 45, 46
- sna.ex, 44
- summary (*summary.zelig*), 45
- summary.zelig, 45
- SupremeCourt, 46
- swiss, 46
- terms.object, 22
- ternaryplot, 47, 49
- ternarypoints, 48, 49
- tobin, 50
- turnout, 50
- user.prompt, 51
- voteincome, 51
- Weimar, 52
- Zelig (*Zelig-package*), 2
- zelig, 36, 37, 40–43, 45, 46, 53, 58
- Zelig-package, 2
- Zelig.url, 54
- zeligDepStatus, 54, 57
- zeligDepUpdate, 55, 56
- zeligDescribeModelXML, 57
- zeligGetSpecial
(*zeligDescribeModelXML*), 57
- zeligInstalledModels
(*zeligDescribeModelXML*), 57
- zeligListModels
(*zeligDescribeModelXML*), 57
- zeligModelDependency
(*zeligDescribeModelXML*), 57