Package ‘Zelig’

February 19, 2015

Maintainer James Honaker <zelig-zee@iq.harvard.edu>

License GPL (>= 2)

Title Everyone's Statistical Software

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

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.

Version 4.2-1

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

Date 2013-09-12

Depends R (>= 2.14), boot, MASS, methods, sandwich

Suggests Amelia, mvtnorm, Formula, gee, survey, survival, systemfit,
       MatchIt, MCMCpack, coda

NeedsCompilation no

Repository CRAN

Date/Publication 2013-09-21 08:14:26
R topics documented:

Zelig-package ................................................. 6
alpha ......................................................... 7
approval ..................................................... 8
as.bootlist ................................................... 8
as.bootvector ............................................... 9
as.data.frame.setx ......................................... 9
as.description ............................................... 10
as.description.description ............................... 11
as.description.list ......................................... 11
as.matrix.pooled.setx ...................................... 12
as.matrix.setx ............................................... 13
as.parameters ............................................... 13
as.parameters.default ..................................... 14
as.parameters.list ......................................... 15
as.parameters.parameters ................................. 16
as.qi .......................................................... 16
as.qi.default ............................................... 17
as.qi.list ................................................... 18
as.qi.qi ....................................................... 18
as.summarized .............................................. 19
as.summarized.list ......................................... 20
as.summarized.summarized.qi ............................ 20
attach.env .................................................... 21
bivariate ...................................................... 22
bootfn.default .............................................. 22
bootstrap ...................................................... 23
bootstrap.default ......................................... 24
bootstrap.gamma ............................................ 24
bootstrap.negbinom ......................................... 25
bootstrap.normal .......................................... 25
callToString .................................................. 26
cite .......................................................... 27
cluster.formula .............................................. 27
cmvglm ....................................................... 28
coalition ...................................................... 28
coalition2 .................................................... 29
coefficients ................................................. 30
combine ...................................................... 30
constructDataFrame ......................................... 31
constructDesignMatrix .................................... 32
depends.on.zelig ........................................... 32
describe ...................................................... 33
describe.default ............................................ 33
describe.exp ................................................ 34
describe.gamma ............................................. 34
describe.logit .............................................. 35
R topics documented:

describe.ls ................................................................. 35
describe.negbinom ..................................................... 36
describe.normal ......................................................... 37
describe.poisson ....................................................... 37
describe.probit ......................................................... 38
describe.tobit ......................................................... 38
describe.zelig ......................................................... 39
description ............................................................... 39
eidat .......................................................... 40
find.match ............................................................. 41
free1 .......................................................... 41
free2 .......................................................... 43
friendship ............................................................ 45
get.package ............................................................ 45
GetObject .............................................................. 46
getPredictorTerms ..................................................... 46
getResponseTerms ..................................................... 47
getResponseTerms.Formula ............................................ 47
getAddressTerms.formula ............................................. 48
getAddressTerms.list ............................................... 49
GetSlot .............................................................. 49
GetSlot.zelig .......................................................... 50
grunfeld ............................................................ 51
has.zelig2 ............................................................. 51
help.zelig ............................................................ 52
hoff ............................................................. 52
homerun ............................................................. 53
ignore ............................................................. 54
immigration ........................................................... 54
is.formula ............................................................. 55
is.qi ............................................................. 55
is.valid.qi.list ........................................................ 56
is.zelig.compliant .................................................... 56
is.zelig.package ....................................................... 57
klein ............................................................. 58
kmenta ............................................................. 58
link ............................................................. 59
linkinv ............................................................. 59
list.depth ............................................................. 60
list.zelig.dependent.packages ....................................... 60
list.zelig.models ..................................................... 61
loadDependencies ..................................................... 61
macro ............................................................. 62
make.parameters ....................................................... 62
makeModelMatrix ....................................................... 63
makeZeligObject ....................................................... 63
MatchIt.url ........................................................... 64
Max ............................................................. 64
<table>
<thead>
<tr>
<th>R topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCMChook</td>
<td>65</td>
</tr>
<tr>
<td>McmcHookFactor</td>
<td>65</td>
</tr>
<tr>
<td>Median</td>
<td>66</td>
</tr>
<tr>
<td>mexico</td>
<td>67</td>
</tr>
<tr>
<td>mi</td>
<td>67</td>
</tr>
<tr>
<td>mid</td>
<td>68</td>
</tr>
<tr>
<td>Min</td>
<td>69</td>
</tr>
<tr>
<td>mix</td>
<td>69</td>
</tr>
<tr>
<td>Mode</td>
<td>70</td>
</tr>
<tr>
<td>model.frame.multiple</td>
<td>70</td>
</tr>
<tr>
<td>model.matrix.multiple</td>
<td>71</td>
</tr>
<tr>
<td>model.matrix.parseFormula</td>
<td>71</td>
</tr>
<tr>
<td>multilevel</td>
<td>72</td>
</tr>
<tr>
<td>name.object</td>
<td>73</td>
</tr>
<tr>
<td>names.qi</td>
<td>74</td>
</tr>
<tr>
<td>newpainters</td>
<td>74</td>
</tr>
<tr>
<td>param</td>
<td>75</td>
</tr>
<tr>
<td>param.default</td>
<td>76</td>
</tr>
<tr>
<td>param.exp</td>
<td>77</td>
</tr>
<tr>
<td>param.gamma</td>
<td>77</td>
</tr>
<tr>
<td>param.logit</td>
<td>78</td>
</tr>
<tr>
<td>param.ls</td>
<td>79</td>
</tr>
<tr>
<td>param.negbinom</td>
<td>79</td>
</tr>
<tr>
<td>param.normal</td>
<td>80</td>
</tr>
<tr>
<td>param.poisson</td>
<td>81</td>
</tr>
<tr>
<td>param.probit</td>
<td>81</td>
</tr>
<tr>
<td>param.relogit</td>
<td>82</td>
</tr>
<tr>
<td>param.relogit2</td>
<td>83</td>
</tr>
<tr>
<td>param.tobit</td>
<td>83</td>
</tr>
<tr>
<td>parameters</td>
<td>84</td>
</tr>
<tr>
<td>parse.formula</td>
<td>85</td>
</tr>
<tr>
<td>parseFormula</td>
<td>85</td>
</tr>
<tr>
<td>parseFormula.Formula</td>
<td>86</td>
</tr>
<tr>
<td>parseFormula.list</td>
<td>86</td>
</tr>
<tr>
<td>PErisk</td>
<td>87</td>
</tr>
<tr>
<td>plot.ci</td>
<td>88</td>
</tr>
<tr>
<td>plot.ML.sim</td>
<td>89</td>
</tr>
<tr>
<td>plot.pooled.sim</td>
<td>90</td>
</tr>
<tr>
<td>plot.sim</td>
<td>91</td>
</tr>
<tr>
<td>plot.simulations</td>
<td>91</td>
</tr>
<tr>
<td>print.qi</td>
<td>92</td>
</tr>
<tr>
<td>print.qi.summarized</td>
<td>92</td>
</tr>
<tr>
<td>print.setx</td>
<td>93</td>
</tr>
<tr>
<td>print.setx.mi</td>
<td>94</td>
</tr>
<tr>
<td>print.sim</td>
<td>94</td>
</tr>
<tr>
<td>print.summary.MCMCZelig</td>
<td>95</td>
</tr>
<tr>
<td>print.summary.pooled.sim</td>
<td>95</td>
</tr>
<tr>
<td>print.summary.relogit</td>
<td>96</td>
</tr>
<tr>
<td>R topics documented:</td>
<td>Page</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>print.summary.relogit2</td>
<td>96</td>
</tr>
<tr>
<td>print.summary.sim</td>
<td>97</td>
</tr>
<tr>
<td>print.summarySim.MI</td>
<td>98</td>
</tr>
<tr>
<td>print.zelig</td>
<td>99</td>
</tr>
<tr>
<td>qi</td>
<td>100</td>
</tr>
<tr>
<td>qi.exp</td>
<td>100</td>
</tr>
<tr>
<td>qi.summarize</td>
<td>100</td>
</tr>
<tr>
<td>reduceMI</td>
<td>101</td>
</tr>
<tr>
<td>relogit</td>
<td>102</td>
</tr>
<tr>
<td>repl</td>
<td>102</td>
</tr>
<tr>
<td>repl.default</td>
<td>103</td>
</tr>
<tr>
<td>repl.sim</td>
<td>103</td>
</tr>
<tr>
<td>replace.call</td>
<td>104</td>
</tr>
<tr>
<td>robust.glm.hook</td>
<td>105</td>
</tr>
<tr>
<td>robust.gee.hook</td>
<td>105</td>
</tr>
<tr>
<td>rocplot</td>
<td>106</td>
</tr>
<tr>
<td>sanction</td>
<td>107</td>
</tr>
<tr>
<td>setx</td>
<td>108</td>
</tr>
<tr>
<td>setx.default</td>
<td>109</td>
</tr>
<tr>
<td>setx.MI</td>
<td>110</td>
</tr>
<tr>
<td>sim</td>
<td>110</td>
</tr>
<tr>
<td>sim.default</td>
<td>112</td>
</tr>
<tr>
<td>sim.MI</td>
<td>113</td>
</tr>
<tr>
<td>simulation.matrix</td>
<td>114</td>
</tr>
<tr>
<td>simulations.parameters</td>
<td>115</td>
</tr>
<tr>
<td>simulations.plot</td>
<td>115</td>
</tr>
<tr>
<td>sna.ex</td>
<td>116</td>
</tr>
<tr>
<td>special_print_LIST</td>
<td>117</td>
</tr>
<tr>
<td>special_print_MATRIX</td>
<td>117</td>
</tr>
<tr>
<td>splitUp</td>
<td>118</td>
</tr>
<tr>
<td>store.object</td>
<td>119</td>
</tr>
<tr>
<td>structuralToReduced</td>
<td>120</td>
</tr>
<tr>
<td>summarize</td>
<td>120</td>
</tr>
<tr>
<td>summarize.default</td>
<td>121</td>
</tr>
<tr>
<td>summary.glm.robust</td>
<td>121</td>
</tr>
<tr>
<td>summary.MI</td>
<td>122</td>
</tr>
<tr>
<td>summary.MI.sim</td>
<td>122</td>
</tr>
<tr>
<td>summary.pooled.sim</td>
<td>123</td>
</tr>
<tr>
<td>summary.Relogit</td>
<td>124</td>
</tr>
<tr>
<td>summary.Relogit2</td>
<td>124</td>
</tr>
<tr>
<td>summary.sim</td>
<td>125</td>
</tr>
<tr>
<td>summary.zelig</td>
<td>125</td>
</tr>
<tr>
<td>SupremeCourt</td>
<td>126</td>
</tr>
<tr>
<td>swiss</td>
<td>126</td>
</tr>
<tr>
<td>t.setx</td>
<td>127</td>
</tr>
<tr>
<td>table.levels</td>
<td>128</td>
</tr>
<tr>
<td>terms.multiple</td>
<td>128</td>
</tr>
<tr>
<td>terms.vglm</td>
<td>129</td>
</tr>
</tbody>
</table>
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: 4.0-11
Date: 2012-10-28
alpha

Depends: R (>= 2.14), boot, MASS, methods, sandwich, survival
Suggests: mvtnorm, Formula
License: GPL version 2 or newer
URL: http://gking.harvard.edu/zelig

Author(s)

Matt Owen <mowen@iq.harvard.edu>, Kosuke Imai, Olivia Lau, and Gary King

See Also

zelig setx sim

| alpha | Extract ancillary parameters from 'parameters' objects |

Description

Extract ancillary parameters from 'parameters' objects

Usage

alpha(param)

Arguments

param a 'parameters' object

Value

the ancillary parameters specified for the statistical model

Author(s)

Matt Owen <mowen@iq.harvard.edu>
**U.S. Presidential Approval Data**

**Description**
Monthy 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

---

**as.bootlist**

*Convert of Vector of Bootstrapped Parameters to a List-style Boot Object*

**Description**
This inverts the “as.bootvector” function, and returns a list containing the slots “alpha” and “beta”.

**Usage**

as.bootlist(bootstraps, lengths, names)

**Arguments**

- **bootstraps** ...
- **lengths** ...
- **names** a character-vector specifying the names of the boot terms

**Value**
...

### as.bootvector

**Convert Boot Object to a Vector**

**Description**

Receives a list with 2 slots as its input, and returns a vector of the two smashed together along with the offsets used to reverse-construct the object.

**Usage**

```r
as.bootvector(obj)
```

**Arguments**

- `obj`: a list with two slots: “alpha” and “beta”. Respectively, these represent bootstrap samples for ancillary parameters and systematic component of the bootstrapped GLM.

**Value**

- a list containing the resulting vector, as well as an object used to reverse-build the list (“obj”) from the resulting call to “bootstrap”.

**Note**

This method is used internally by Zelig to allow an intuitive, “param”-like API for bootstrapping.

**Author(s)**

Matt Owen <mowen@iq.harvard.edu>

---

### as.data.frame.setx

**Coerce a setx Object into a data.frame**

**Description**

Coerce a setx Object into a data.frame

**Usage**

```r
## S3 method for class 'setx'
as.data.frame(x, row.names=NULL, optional=FALSE, ...)
```
Arguments

- `x` a `setx` object
- `row.names` ignored parameter
- `optional` ignored parameter
- `...` ignored parameters

Value

The `setx` object interpreted as a `data.frame`. The column-names of the resulting `data.frame` are specified by the names of the `setx` object. The row-names are typically unlabeled.

Note

In subsequent versions of Zelig, this version is expected to undergo minor modifications.

### as.description

**Generic Method for Casting 'description' Objects**

**Description**

Convert the result of a call to the 'describe' method into an object parseable by Zelig. Currently conversions only exist for lists and description objects.

**Usage**

```r
as.description(descr, ...)
```

**Arguments**

- `descr` an object to cast an object of type 'description'
- `...` parameters which are reserved for future Zelig revisions

**Value**

an object of type 'description'

**Author(s)**

Matt Owen <mowen@iq.harvard.edu>

**See Also**

- `as.description.description`  
- `as.description.list`
as.description.description

\textit{description -> description}

\textbf{Description}

Identity operation on a description object.

\textbf{Usage}

\begin{verbatim}
## S3 method for class 'description'
as.description(descr, ...)
\end{verbatim}

\textbf{Arguments}

\begin{itemize}
  \item descr an object of type 'description'
  \item ... ignored
\end{itemize}

\textbf{Value}

the same object

\textbf{Author(s)}

Matt Owen <mowen@iq.harvard.edu>

\hline

as.description.list \textit{list -> description}

\textbf{Description}

Convert list into a description object.

\textbf{Usage}

\begin{verbatim}
## S3 method for class 'list'
as.description(descr, ...)
\end{verbatim}

\textbf{Arguments}

\begin{itemize}
  \item descr a list
  \item ... ignored
\end{itemize}

\textbf{Value}

an object of type 'description'
as.matrix.pooled.setx

Convert a “pooled.setx” Object to a Matrix

Description

The setx object is, in its most basic form, a list of column names and values specified for each of these column names. This function simply converts the key-value pairs of column-name and specified value into a matrix.

Usage

```r
## S3 method for class 'pooled.setx'
as.matrix(x, ...)
```

Arguments

- `x` a setx object
- `...` ignored parameters

Value

a matrix containing columns and rows corresponding to the explanatory variables specified in the call to the ‘setx’ function

Note

This method allows basic matrix arithmetic operations on data objects, which mirror values stored within setx objects. In many scenarios, simulations require matrix-multiplication, etc. to be performed on a data-set. This function facilitates that need.

Author(s)

Matt Owen <mowen@iq.harvard.edu>
as.matrix.setx

Convert a 'setx' Object to a Matrix

Description
The setx object is, in its most basic form, a list of column names and values specified for each of these column names. This function simply converts the key-value pairs of column-name and specified value into a matrix.

Usage

```r
## S3 method for class 'setx'
as.matrix(x, ...)
```

Arguments

- `x`: a setx object
- `...`: ignored parameters

Value

A matrix containing columns and rows corresponding to the explanatory variables specified in the call to the 'setx' function.

Note

This method allows basic matrix arithmetic operations on data objects, which mirror values stored within setx objects. In many scenarios, simulations require matrix-multiplication, etc. to be performed on a data-set. This function facilitates that need.

Author(s)

Matt Owen <mowen@iq.harvard.edu>

as.parameters

Generic Method for Converting Objects into 'parameters'

Description

Converts list-style objects into Parameter lists primarily used by the ‘qi’ methods. These list-style objects may contain keys specifying: 'link' (the link function of a statistical model), 'linkinv' (the inverse-link function), 'family' (a object of ‘family’ class used to specify the model’s classification), 'alpha' (a vector of ancillary parameters, and 'simulations' (a vector of simulated draws from the model’s underlying distribution.

as.parameters.default

Usage

    as.parameters.default(params, num = NULL, ...)

Arguments

    params          the object to be casted
    num             an integer specifying the number of simulations to compute
    ...             ignored

Value

    the object passed in

Note

    Only three scenarios may exist - converting 'parameters' to 'parameters', 'list' to 'parameters', and
    vectors to 'parameters'. The third in particular is needed only for backwards compatibility, and
    support will likely be deprecated.
    Furthermore, this function should be exclusively used implicitly and by Zelig.

Author(s)

    Matt Owen <mowen@ig.harvard.edu>

See Also

    as.parameters.list as.parameters.parameters, as.parameters.default

Description

    ??? -> parameters

Usage

    as.parameters.default(params, num = NULL, ...)

Arguments

    params          any non-supported data-type
    num             an integer specifying the number of simulations to compute
    ...             ignored

Value

    the object passed in
as.parameters.list  

Note
This function should be deprecated.

Author(s)
Matt Owen <mowen@iq.harvard.edu>

as.parameters.list  list -> parameters

Description
The list may contain: 'link', 'linkinv', 'family', 'alpha', and 'simulations' keys.

Usage
as.parameters.list(params, num = NULL, ...)

Arguments
params a list object
num an integer specifying the number of simulations to be taken
... ignored parameters

Value
an object of type 'parameters'

Author(s)
Matt Owen <mowen@iq.harvard.edu>

See Also
as.parameters
**Description**

parameters -> parameters This is merely an identity function when casting 'parameters' objects into 'parameters'.

**Usage**

as.parameters.parameters(params, ...)

**Arguments**

- **params**: a parameters object
- **...**: ignored parameters

**Value**

the same parameter object

**Author(s)**

Matt Owen <mowen@iq.harvard.edu>

---

**as.qi**

Generic Method for Converting Various Objects into 'qi' Objects 'qi' objects are list-style objects used by the 'summarize' function to compute simple summaries about the simulated data. For readability and simplicity purposes, the 'qi' function typically returns a list of named simulations. This list is converted internally by Zelig into a 'qi' object so that several methods can be easily applied to the Quantities of Interest: plot, summarize, and print

**Description**

Generic Method for Converting Various Objects into 'qi' Objects 'qi' objects are list-style objects used by the 'summarize' function to compute simple summaries about the simulated data. For readability and and simplicity purposes, the 'qi' function typically returns a list of named simulations. This list is converted internally by Zelig into a 'qi' object so that several methods can be easily applied to the Quantities of Interest: plot, summarize, and print
as.qi.default

Usage
   as.qi(s)

Arguments
   s  the object to be casted

Value
   an object of type ‘qi’

Note
   These functions are primarily used internally by Zelig and should not be used in the Global namespace.

Author(s)
   Matt Owen <mowen@iq.harvard.edu>

See Also
   as.qi.default as.qi.qi as.qi.list

Description
   ??? -> qi

Usage
   as.qi.default(s)

Arguments
   s  any unsupported object

Value
   an object of type ‘qi’

Author(s)
   Matt Owen <mowen@iq.harvard.edu>
as.qi.list

Description

list -> qi This function has a lot of room to go wrong. It tries to detect whether the zelig model is old-style or new-style (as of 4/4/2011). Eventually this feature should be phased out.

Usage

as.qi.list(s)

Arguments

s a list

Value

an object of type ‘qi’

Note

This method has peculiar behavior when the list contains only two elements. The crucial fix is to simply remove the portion of code which intentionally implements this peculiar behavior.

Author(s)

Matt Owen <mowen@iq.harvard.edu>

as.qi.qi

qi -> qi

Description

qi -> qi

Usage

as.qi.qi(s)

Arguments

s an object of type ‘qi’
as.summarized

Value

s an object of type ‘qi’

Author(s)

Matt Owen <mowen@iq.harvard.edu>

---

as.summarized  
Generic Method for Casting Objectst as 'summarized' Objects

Description

This function is particularly for use by the 'summarize' method, which summarizes the simulations taken from the 'qi' method. The generic function 'summary' when applied to a Zelig Simulation implicitly uses this function.

Usage

as.summarized(x, ...)

Arguments

x  an object

...  unspecified parameters

Value

a 'summarized.qi' object

Note

This is made available on the Global namespace as a matter of potential future compliance.

Author(s)

Matt Owen <mowen@iq.harvard.edu>
as.summarized.list  
list -> summarized.qi Convert a list into a “summarized.qi” object

Description

list -> summarized.qi Convert a list into a “summarized.qi” object

Usage

```r
## S3 method for class 'list'
as.summarized(x, ...)
```

Arguments

- `x` a list
- `...` ignored parameters

Value

a “summarized.qi” object

Author(s)

Matt Owen <mowen@iq.harvard.edu>

---

as.summarized.summarized.qi

summarized.qi -> summarized.qi

Description

Identity operation on “summarized.qi” objects

Usage

```r
## S3 method for class 'summarized.qi'
as.summarized(x, ...)
```

Arguments

- `x` an object of type 'summarized.qi'
- `...` ignored parameters

Value

the same 'summarized.qi’ object
**attach.env**  

**Author(s)**  
Matt Owen <mowen@iq.harvard.edu>

---

**Description**  
Returns a function, specified by the user, with the variables of a specified environment attached. This, in essence, allows programmers to write functions that have forms of private memory. This makes the function behave similarly to an object.

**Usage**  
```
attach.env(f, env = NULL, ...)
```

**Arguments**  

- `f` a function which will be modified  
- `env` an environment variable which will be attached to the function being returned  
- `...` arbitrary key-value paired parameters which will be assigned to the environment of the function being returned

**Value**  
the original function “f” with a different environment attached to it.

**Note**  
This function is used by Zelig to ensure that particular method calls - param, qi, bootstrap - will contain the private variables: “.fitted”, “.model”, “.call” and “.env” which respectively contain the fitted model object, the name of the zelig model being invoked, the original call to the model-fitting function and the environment in which to call the function call.

**Author(s)**  
Matt Owen <mowen@iq.harvard.edu>
Sample data for bivariate probit regression

**Description**
Sample data for the bivariate probit regression.

**Usage**
```r
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**

---

Default Boot-strapping procedure

**Description**
The default procedure for extracting bootstrap information. Note that this method re-fits the data and resamples the data frequently. This is a good candidate for fixing-up.

**Usage**
```r
bootfn.default(data, i, object, bootstrapfn = NULL, num, ...)  
```

**Arguments**
- `data` a data.frame
- `i` an integer or character-string specifying the index of the row to be used in the bootstrapping procedure.
- `object` the fitted model object
- `bootstrapfn` a function used to bootstrap the object
- `num` an integer specifying the number of samples to simulate
- `...` unspecified parameters
bootstrap

Value

a list of parameters

Description

This method is intended to be overried by statistical models that would like to support statistical bootstrapping.

Usage

bootstrap(obj, ...)

Arguments

obj a fitted model object that will be used to produce boot-strapped parameters. This object usually inherits the class “glm” or “lm” object

... unspecified parameters

Value

a list with the “alpha” and “beta” slots set. Note that “alpha” corresponds to ancillary parameters and “beta” corresponds to systematic components of the model

Note

This method has private memory storage and can reference the objects: “.fitted”, “.data”, “.call”, “.env”, despite having no declaration in the argument list.

Author(s)

Matt Owen <mowen@iq.harvard.edu>


**bootstrap.default**

*Produce Boot-strapped Parameters for a Statistical Model*

**Description**

This method is a fallback for bootstrapping models that do not have a defined “bootstrap” method. For most models, this default is sufficient, so long as the model follows the usual convention that “coef(obj)” returns the systematic parameters of a fitted model.

**Usage**

```r
## Default S3 method:
bootstrap(obj, ...)
```

**Arguments**

- `obj`: a fitted model object. This is typically of type “glm” or “lm”
- `...`: unspecified parameters

**Value**

a list with the “alpha” and “beta” slots set

**Author(s)**

Matt Owen <mowen@iq.harvard.edu>

---

**bootstrap.gamma**

*Bootstrap Parameters for Zelig “gamma” GLM*

**Description**

Returns bootstrapped parameter estimates for a “gamma” GLM.

**Usage**

```r
## S3 method for class 'gamma'
bootstrap(obj, ...)
```

**Arguments**

- `obj`: a “zelig” object that will be used to produce boot-strapped parameters
- `...`: extra parameters to be passed to the “boot” method. These are typically ignored, but is included for further expansion.
**bootstrap.negbinom**

**Value**

a list containing information concerning link, link-inverses, etc.

**Author(s)**

Matt Owen <mowen@iq.harvard.edu>

---

**bootstrap.normal**

*Bootstrap Parameters for Zelig “normal” GLM*

**Description**

Returns bootstrapped parameter estimates for a Gaussian GLM.

**Usage**

```r
## S3 method for class 'normal'
bootstrap(obj, num, ...)
```

**Arguments**

- `obj`: a “zelig” object that will be used to produce boot-strapped parameters
- `...`: extra parameters to be passed to the “boot” method. These are typically ignored, but is included for further expansion.

**Value**

a list containing information concerning link, link-inverses, etc.

**Author(s)**

Matt Owen <mowen@iq.harvard.edu>
Arguments

- **obj**  
  A “zelig” object that will be used to produce boot-strapped parameters
- **num**  
  An integer specifying the number of simulations to produce
- **...**  
  Extra parameters to be passed to the “boot” method. These are typically ignored, but is included for further expansion.

Value

A list containing information concerning link, link-inverses, etc.

Author(s)

Matt Owen <mowen@iq.harvard.edu>

---

callToString  
Convert call Object to a String

Description

This method converts call objects into a simple, intuitive human-readable form.

Usage

callToString(x, ...)  

Arguments

- **x**  
  A call object
- **...**  
  Ignored parameters

Value

A character-string representing the call object

Author(s)

Matt Owen <mowen@iq.harvard.edu>
Citation information for a 'description' object

Description
Citation information for a 'description' object

Usage
cite(descr)

Arguments
descr an object of type 'description'

Value
a character-string giving citation info

Author(s)
Matt Owen <mowen@iq.harvard.edu>

Generate Formulae that Consider Clustering

Description
This method is used internally by the "Zelig" Package to interpret clustering.

Usage
cluster.formula(formula, cluster)

Arguments
formula a formula object
cluster a vector

Value
a formula object describing clustering
cmvglm

Description

cmvglm

Usage

cmvglm(formula, model, ndim, data = NULL, fact = NULL)

Arguments

  formula  a formula
  model    the names of the Zelig model
  ndim     the number of dimensions in the statistical model
  data     a data-frame
  fact     ???

Author(s)

  Kosuke Imai and Olivia Lau

coalition

Coalition Dissolution in Parliamentary Democracies

Description

This data set contains survival data on government coalitions in parliamentary democracies (Bel-
gium, Canada, Denmark, Finland, France, Iceland, Ireland, Israel, Italy, Netherlands, Norway, Por-
tugal, 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


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

coirition2

_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_


_Source_

ICPSR

_References_


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

Description

Returns simulated parameters of coefficients for use in statistical simulation. The values are set by the model-fitting function and the developer of the qi.<model name> method.

Usage

```r
## S3 method for class 'parameters'
coef(object, ...)
```

Arguments

- `object`: a 'parameters' object
- `...`: ignored

Value

simulations, specified by the Zelig model, of the ancillary parameters

Note

This function may not differ at all from coef.default

Author(s)

Matt Owen <mowen@iq.harvard.edu>

---

combine  

Description

Produce All Combinations of a Set of Lists

Usage

```r
combine(...)```

Arguments

- `...`: a set of lists to mix together
Value

all the combinations of the lists with repetition

Note

This function is used internally by the ‘mi’ constructors in order to produce the complete set of combinations of data-frames and factors by to subset the data-frames.

Author(s)

Matt Owen <mowen@iq.harvard.edu>

---

**constructDataFrame**

*Construct Data Frame* Construct and return a tiny (single-row) data-frame from a larger data-frame, a list of specified values, and a formula

---

**Description**

Construct Data Frame Construct and return a tiny (single-row) data-frame from a larger data-frame, a list of specified values, and a formula

**Usage**

`constructDataFrame(data, specified)`

**Arguments**

- `data`: a “data.frame” that will be used to create a small design matrix
- `specified`: a list with key-value pairs that will be used to explicitly set several values

**Value**

a “data.frame” containing a single row
**constructDesignMatrix**  
*Construct Design Matrix from Construct and return a design matrix based on a tiny data-frame (single-row).*

**Description**  
Construct Design Matrix from Construct and return a design matrix based on a tiny data-frame (single-row).

**Usage**  
constructDesignMatrix(data, formula)

**Arguments**
- **data**  
a “data.frame” (preferably single-rowed) that will be used to create a small design matrix
- **formula**  
a formula, whose predictor variables will be used to create a design matrix

**Value**  
a design (model) matrix

---

**depends.on.zelig**  
*Whether a Statistical Package Depends on the Zelig Software Suite*

**Description**  
Whether a Statistical Package Depends on the Zelig Software Suite

**Usage**  
depends.on.zelig(package = "")

**Arguments**
- **package**  
a character-string representing a package name

**Value**  
whether the package lists Zelig as a dependency in its DESCRIPTION

**Note**  
This function is used primarily internally to determine whether a package is contributing a function to the Zelig software suite
**describe**  
*Method to describe a model to Zelig*

**Description**
Method to describe a model to Zelig

**Usage**
```r
describe(...)```

**Arguments**

```r
... parameters which are typically ignored```

**Value**
a list to be processed by `as.description`

**Author(s)**
Matt Owen <mowen@iq.harvard.edu>

---

**describe.default**  
*Default describe function for an arbitrary model This method exists solely as a backup when an author does not contribute a 'describe' function for their model*

**Description**
Default describe function for an arbitrary model This method exists solely as a backup when an author does not contribute a 'describe' function for their model

**Usage**
```r
## Default S3 method:
describe(...)```

**Arguments**

```r
... dummy parameters purely to cast the correct object. That is, the parameters of
the function should not BE referenced specifically```

**Value**
a list to be processed by `as.description`
describe.exp

Describe a “exp” model to Zelig

Description
Describe a “exp” model to Zelig

Usage

```r
## S3 method for class 'exp'
describe(...)```

Arguments

... ignored parameters

Value

a list to be processed by `as.description`

Author(s)

Matt Owen <mowen@iq.harvard.edu>

describe.gamma

Describe the gamma model to Zelig

Description
Describe the gamma model to Zelig

Usage

```r
## S3 method for class 'gamma'
describe(...)```

Arguments

... ignored parameters

Value

a list of important information

Author(s)

Matt Owen <mowen@iq.harvard.edu>
describe.logit  

Author(s)  
Matt Owen <mowen@iq.harvard.edu>  

---  

**describe.logit**  
*Describe a ‘logit’ model to Zelig*  

---  

**Description**  
Describe a ‘logit’ model to Zelig  

**Usage**  
```r  
## S3 method for class 'logit'
describe(...)  
```

**Arguments**  
... ignored parameters  

**Value**  
a list to be processed by ‘as.description’  

**Author(s)**  
Matt Owen <mowen@iq.harvard.edu>  

---  

**describe.lm**  
*Describe a lm model to Zelig*  

---  

**Description**  
Describe a lm model to Zelig  

**Usage**  
```r  
## S3 method for class 'lm'
describe(...)  
```

**Arguments**  
... ignored parameters  

**Value**  
a list to be processed by as.description
Note

ls stands for "least squares fit"

Author(s)

Matt Owen <mowen@iq.harvard.edu>

describe.negbinom  

Describe the negbinom model to Zelig

Description

Describe the negbinom model to Zelig

Usage

## S3 method for class 'negbinom'
describe(...)  

Arguments

... ignored parameters

Value

a list to be processed by as.description

Note

negbinom stands for "negative binomial"

Author(s)

Matt Owen <mowen@iq.harvard.edu>
describe.normal  

*Description*

Describe the normal model to Zelig

*Usage*

```r
## S3 method for class 'normal'
describe(...)```

*Arguments*

... ignored parameters

*Value*

a list to be processed by `as.description`

*Author(s)*

Matt Owen <mowen@iq.harvard.edu>

---

describe.poisson  

*Description*

Describe the `poisson` model to Zelig

*Usage*

```r
## S3 method for class 'poisson'
describe(...)```

*Arguments*

... ignored parameters

*Value*

a list to be processed by `as.description`

*Author(s)*

Matt Owen <mowen@iq.harvard.edu>
describe.probit  

Describe the ‘probit’ model to Zelig

Description

Describe the ‘probit’ model to Zelig

Usage

```r
## S3 method for class 'probit'
describe(...)
```

Arguments

... ignored parameters

Value

a list to be processed by ‘as.description’

Author(s)

Matt Owen <mowen@iq.harvard.edu>

describe.tobit  

Describe a “tobit” model to Zelig

Description

Describe a “tobit” model to Zelig

Usage

```r
## S3 method for class 'tobit'
describe(...)
```

Arguments

... ignored parameters

Value

a list to be processed by ‘as.description’

Author(s)

Matt Owen <mowen@iq.harvard.edu>
describe.zelig

Get Description Object Used to Cite this Zelig Model

Description

Get Description Object Used to Cite this Zelig Model

Usage

```r
## S3 method for class 'zelig'
describe(object, ...)
```

Arguments

- `object`: a `zelig` object
- `...`: ignored parameters

Value

- a `description` object used internally to produce citation text

Note

This function should be reevaluated in design, since `description` objects are exclusively used internally. In particular, this method would be more useful to users as a `cite` method.

Author(s)

Matt Owen <mowen@iq.harvard.edu>

description

Constructor for the `description` class

Description

Constructor for the `description` class

Usage

```r
description(authors = c("Kosuke Imai", "Gary King", "Olivia Lau"),
  year = NULL, model = "", text = "", url = "",
category = NULL)
```
**Arguments**

- **authors**: a character-vector of author names
- **year**: a numeric specifying the year
- **model**: a character-string specifying model name
- **text**: a character-string specifying the title of the model. This typically includes more exact information than 'model'. E.g., for the 'logit' the title 'Logistic Regression for Dichotomous Variables' would be a suitable text parameter.
- **url**: a character-string specifying the model’s software page
- **category**: deprecated until data-verse bindings are reevaluated

**Value**

- an object of type 'description'

**Author(s)**

Matt Owen <mowen@iq.harvard.edu>

---

**Description**

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

**Usage**

```r
data(eidat)
```

**Format**

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

*Find a Partial or Exact Match from a Vector of Strings Searches a vector of character-string, and returns the best match.*

**Description**

Find a Partial or Exact Match from a Vector of Strings Searches a vector of character-string, and returns the best match.

**Usage**

```r
find.match(needle, haystack, fail = NA)
```

**Arguments**

- `needle` a character-string to search for in the
- `haystack` a vector of character-strings
- `fail` the value to return in case no match is found. Defaults to NA

**Details**

“find.match” attempts to use several common matching functions in an order that sequentially prefers less strict matching, until a suitable match is found. If none is found, then return the value of the “fail” parameter (defaults to NA). The functions used for matching are: “match”, “charmatch”, and finally “grep”.

**Value**

the best-matched string or NA

**Author(s)**

Matt Owen <mowen@iq.harvard.edu>

---

**free1**

*Freedom of Speech Data*

**Description**

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

**Usage**

```r
data(free1)
```
Details

A table with 150 observations and 12 variables.

- sex1 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
- 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


---

**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.

- sex1 for men and 0 for women
- ageAge of respondent in years
- educLevels of education, coded as a numeric variable with
  - 1No formal education
  - 2Less than primary school education
  - 3Completed primary school
  - 4Completed secondary school
  - 5Completed high school
  - 6Completed college
  - 7Completed post-graduate degree
• 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


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, per)

## End(Not run)

get.package

Find the Zelig package that a particular model belong to

Description

This method is used to help transition Zelig v3.5 users to Zelig v4

Usage

get.package(model, quiet = TRUE, ...)

Arguments

model a character-string specifying a Zelig model
quiet a logical indicating whether to display messages and warnings
... ignored parameters
Value
NA or a character-string specifying the name of the package which contains a specific model

Author(s)
Matt Owen <mowen@iq.harvard.edu>

---

**GetObject**

*Extract the fitted model object from the Zelig object*

---

**Description**
Extract the fitted model object from the Zelig object

**Usage**

GetObject(obj)

**Arguments**

`obj` an object of type ‘zelig’

**Value**
the fitted model object

**Author(s)**
Matt Owen <mowen@iq.harvard.edu>

---

**getPredictorTerms**

*Get Predictor Terms from Zelig-style Formulae*

---

**Description**
This function extracts the predictor terms from a Zelig-style object.

**Usage**

getPredictorTerms(x, ...)

**Arguments**

`x` a Zelig-style formula (‘formula’ or ‘list’)

`...` ignored parameters
getResponseTerms

Value

a character-vector or NA

Note

This function is used exclusively in the development of Zelig-core.

Author(s)

Matt Owen <mowen@iq.harvard.edu>

ggetResponseTerms

Get Response Terms from a Zelig-style Formula

Description

This method acquires the response variables from Zelig-style input.

Usage

getResponseTerms(x, ...)

Arguments

x a formula or list of formulae
... ignored parameters

Value

a character-vector specifying the response terms in this formula

ggetResponseTerms.Formula

Get Response Terms from a “Formula” Object

Description

This method gets the response terms from a “Formula” Object

Usage

## S3 method for class ‘Formula’
ggetResponseTerms(x, ...,
    single.only=FALSE, duplicates=TRUE)
getResponseTerms.formula

Arguments

x a formula
...

single.only a logical specifying whether 'cbind' or 'list' keywords are allowed

duplicates a logical specifying whether the returned character-vector will only return duplicates.

Value

a character-vector specifying the response terms of the formula

Author(s)

Matt Owen

getDescription

Get Response Terms from a Standard Formula

Description

This method gets the response terms from a standard formula

Usage

## S3 method for class 'formula'
getResponseTerms(x, ..., single.only=FALSE, duplicates=TRUE)

Arguments

x a formula
...

single.only a logical specifying whether 'cbind' or 'list' keywords are allowed

duplicates a logical specifying whether the returned character-vector will only return duplicates.

Value

a character-vector specifying the response terms of the formula

Author(s)

Matt Owen
getResponseTerms.list  Get Response Terms from a List-style Formula

Description

This method gets the response terms from a standard formula.

Usage

```r
## S3 method for class 'list'
getResponseTerms(x, ...)
```

Arguments

- `x` a list of formulae
- `...` ignored parameters

Value

a character-vector specifying the response terms of the formula.

Author(s)

Matt Owen

GetSlot  Generic method for extracting variables from both S3 and S4 fitted model object

Description

Generic method for extracting variables from both S3 and S4 fitted model object.

Usage

```r
GetSlot(obj, key, ...)
```

Arguments

- `obj` an object of type 'zelig'
- `key` a character-string specifying the name of the variable to extract
- `...` typically ignored parameters

Value

the value of that extracted object or NULL.
GetSlot.zelig  

Return a Value from a zelig Fitted Model

Description

Returns a value from the result of a model fitting function

Usage

```r
## S3 method for class 'zelig'
GetSlot(obj, key, ...)
```

Arguments

- `obj`  
  a zelig object

- `key`  
  a character-string specifying the which value to extract from the fitted model object

- `...`  
  subsequent values to extract from the fitted model object

Value

values of the specified keys

Note

This function is primarily used by Zelig developers within q1 functions

Author(s)

Matt Owen mowen@iq.harvard.edu
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.

has.zelig2

Whether an Arbitrary R-package has a Zelig2 Function within Its Namespace

Description

Whether an Arbitrary R-package has a Zelig2 Function within Its Namespace

Usage

has.zelig2(pkg)

Arguments

pkg a character-string representing a package name

Value

whether the package contains any zelig2-functions

Note

This function is used primarily internally to determine whether a a package is contributing a function to the Zelig software suite
help.zelig

Help system for Zelig models

Description
Help system for Zelig models

Usage
help.zelig(...)

Arguments
... the help files to look-up

Value
results of calling the specific help function

Author(s)
Matt Owen mowen@iq.harvard.edu

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


---

**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

```r
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

ignore

Constructor for the ‘ignore’ class This class is included for future use, and is currently not used in any Zelig model. It is designed for use with zelig2* functions

Description

Constructor for the ‘ignore’ class This class is included for future use, and is currently not used in any Zelig model. It is designed for use with zelig2* functions

Usage

ignore(default = NULL, type = "no pass")

Arguments

default default value
type ignored parameter

Value

an ‘ignore’ object

Author(s)

Matt Owen mowen@iq.harvard.edu

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


Source

National Election Survey
is.formula

Whether an Object is a Formula

Description
This is a boolean-check to see whether an object is a formula.

Usage
is.formula(x)

Arguments

x an object

Value

a logical specifying whether an object is a formula

Note
This will not be shared in the Zelig/ZeligFormulae namespace.

Author(s)
Matt Owen

is.qi

Test If Value is Interpretable as a QI

Description
Test If Value is Interpretable as a QI

Usage
is.qi(qi)

Arguments

qi a potential quantity of interest
Value

A logical specifying whether this value should or should-not be output

Author(s)

Matt Owen <mowen@iq.harvard.edu>

is.valid.qi.list  Check If Object Is a List of Valid Quantities of Interest

Description

Check If Object Is a List of Valid Quantities of Interest

Usage

is.valid.qi.list(x)

Arguments

x  an object to be tested

Value

TRUE or FALSE

Author(s)

Matt Owen <mowen@iq.harvard.edu>

is.zelig.compliant  Whether a R-Package Contains a 'Yes' in its DESCRIPTION File's 'Zelig' Field

Description

Whether a R-Package Contains a 'Yes' in its DESCRIPTION File's 'Zelig' Field

Usage

is.zelig.compliant(package = "")

Arguments

package  a character-string specifying an installed R-package
Value

whether the package’s DESCRIPTION file specifies Zelig-compliancy

Note

This package was used internally to determine whether an R-package is Zelig compliant, but is now likely deprecated.

Author(s)

Matt Owen <mowen@iq.harvard.edu>

See Also

is.zelig.package

Description

Whether an Installed R-Pack Depends on Zelig

Usage

is.zelig.package(package = "")

Arguments

package a character-string naming a package

Value

whether this package depends on Zelig

Note

This package was used internally to determine whether an R-package is Zelig compliant, but is now likely deprecated. This test is useless if not paired with
### 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**

```r
data(klein)
```

**Format**


**Source**

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

### kmenta

**Simulation Data for model Three-Stage Least Square (threesls) that corresponds to method 3SLS of systemfit**

**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**

```r
data(kmenta)
```

**Format**

A table containing 5 variables ("q", "p", "d", "f", "a") and 20 observations.
Method for extracting the link function from 'parameters' objects

**Description**
Method for extracting the link function from 'parameters' objects

**Usage**
link(param)

**Arguments**
- param: a 'parameters' object

**Value**
the link function specified by the 'param' function for the given Zelig model

**Author(s)**
Matt Owen <mowen@iq.harvard.edu>

Method for extracting the inverse link function from 'parameters' objects

**Description**
Returns the inverse link function of a “parameters” object. If the model’s developer did not specify one (but did specify a link function) this function returns a numerical approximation of the link function.

**Usage**
linkinv(param)

**Arguments**
- param: a 'parameters' object

**Value**
the inverse link function specified by the 'param' function for the given Zelig model

**Author(s)**
Matt Owen <mowen@iq.harvard.edu>
### list.depth

*Count the Depth of a List Object*

**Description**

This function recursively computes the depth of a list object. That is, it determines how many layers or levels exist within the object.

**Usage**

```r
list.depth(obj)
```

**Arguments**

- `obj` a vector or list object

**Note**

This function is used internally by Zelig.

**Author(s)**

Matt Owen <mowen@iq.harvard.edu>

### list.zelig.dependent.packages

*Get a List of Packages Installed on the Current Machine that Depend on Zelig*

**Description**

Get a List of Packages Installed on the Current Machine that Depend on Zelig

**Usage**

```r
list.zelig.dependent.packages()
```

**Value**

a character-vector of all zelig-dependent packages on the current machine

**Note**

This function is used primarily internally to determine whether a package is contributing a function to the Zelig software suite


**list.zelig.models**  
*List Zelig Models Installed on the Current Machine*

**Description**  
List Zelig Models Installed on the Current Machine

**Usage**  
```
list.zelig.models(with.namespace = TRUE)
```

**Arguments**  
- `with.namespace`  
a boolean specifying whether

**Value**  
list of all zelig models

**Note**  
This list is not necessarily complete

**loadDependencies**  
*Load External Dependencies Safely and Dynamically*

**Description**  
“loadDependencies” is a helper function for loading external dependencies at runtime.

**Usage**  
```
loadDependencies(..., character.only = FALSE)
```

**Arguments**  
- `...`  
A collection of packages to load. If “character.only”=FALSE, these can be entered symbolically (e.g. `loadDependencies(MASS)`). Otherwise, these arguments are character-strings.
- `character.only`  
a boolean specifying whether the arguments are strictly character-strings.

**Value**  
TRUE (invisibly) if successful. Otherwise the script is stopped.

**Note**  
This is used by Zelig developers to dynamically load “dependent” packages at runtime.
make.parameters

Description

??? For use with cmvglm

Usage

make.parameters(terms, shape = "vector",
ancillary = TRUE, eqns = NULL)

Arguments

terms ???
shape ???
ancillary ???
eqns ???

make.parameters

Macroeconomic Data

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.
Value

???

Author(s)

Kosuke Imai and Olivia Lau

---

**makeModelMatrix**  
*Make a Model Matrix from a Zelig-Style Formula*

**Description**

This is a helper function that creates a model.matrix like object of Zelig-style formulae.

**Usage**

```r
makeModelMatrix(formula, data)
```

**Arguments**

- `formula`: a Zelig-style formula
- `data`: a data.frame

**Value**

a design (or model) matrix

**Author(s)**

Matt Owen <mowen@iq.harvard.edu>

---

**makeZeligObject**  
*Make an Individual Zelig Object*

**Description**

Returns a “zelig” object with the proper specifications

**Usage**

```r
makeZeligObject(object, model, call, zelig_call, data, label, env, package.name = NULL)
```
Arguments

object a fitted statistical model
model a character-string specifying the name of the model
call The call that produced the fitted model
zelig_call The call made to the original zelig function
data the data.frame used to fit the model
label a character-string or symbol used as a human-readable label for the data-set
eenv an environment variable that contains all variables to evaluate the call “zelig_call”
package.name a character-string specifying the name of the package that is the source of the model used to fit this object

Value

A “zelig” object

Description

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

Max

Compute the Maximum Value of a Vector

Description

Compute the Maximum Value of a Vector

Usage

Max(x, na.rm = NULL)

Arguments

x a numeric or ordered vector
na.rm ignored

Value

the maximum value of the vector

Author(s)

Matt Owen <mowen@iq.harvard.edu>
MCMChook

*Hook to Clean-up MCMC Objects*

**Description**

This method gives valid methods to the resulting MCMC object so that it can be used with Zelig.

**Usage**

```r
MCMChook(obj, model.call, zelig.call, seed = NULL, ..., data = NULL)
```

**Arguments**

- `obj`: the fitted model object (in this case a `mcmc` object).
- `model.call`: the call made to the external model
- `zelig.call`: the actual call to zelig itself
- `seed`: a seed for the MCMC algorithm
- `...`: ignored parameters
- `data`: the data.frame being used to fit the statistical model

**Value**

an object useable by Zelig

**Note**

This function is used internally by the ZeligBayesian package.

**Author(s)**

Olivia Lau, Kosuke Imai, Gary King and Matt Owen

---

McmcHookFactor

*Hook to Clean-up MCMC Factor Object*

**Description**

This method gives valid methods to the resulting MCMC object so that it can be used with Zelig.

**Usage**

```r
McmcHookFactor(obj, model.call, zelig.call, seed = NULL, 
                ...)
Arguments

- **obj**: the fitted model object (in this case a `mcmc` object.
- **model.call**: the call made to the external model
- **zelig.call**: the actual call to zelig itself
- **seed**: a seed for the MCMC algorithm
- **...**: ignored parameters

Value

an object useable by Zelig

Note

This function is used internally by the ZeligBayesian package.

Author(s)

Olivia Lau, Kosuke Imai, Gary King and Matt Owen

**Median**

*Compute the Statistical Median of a Vector*

Description

Compute the Statistical Median of a Vector

Usage

```
Median(x, na.rm = NULL)
```

Arguments

- **x**: a vector of numeric or ordered values
- **na.rm**: ignored

Value

the median of the vector

Author(s)

Matt Owen <mowen@iq.harvard.edu>
**mexico**

*Voting Data from the 1988 Mexican Presidential Election*

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

**Usage**
```r
data(mexico)
```

**Format**
A table containing 33 variables and 1,359 observations.

**Source**
ICPSR

**References**

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

---

**mi**

*Bundle Data-sets for Multiple Imputation*

**Description**
This object prepares data-sets for processing with multiple imputation.

**Usage**
```r
mi(...)```

**Arguments**

*...*

a set of *data.frame’s*

**Value**

an *almost.mi* object, which contains the important internals of a valid, useful *mi* object
Note

This function is largely identical to simply creating a list object, with the exception that any un-named data-sets are automatically labeled via the substitute function

Author(s)

Matt Owen <mowen@iq.harvard.edu>

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


Min

Compute the Minimum Value of a Vector

Description

Compute the Minimum Value of a Vector

Usage

\[
\text{min}(x, \text{na.rm} = \text{NULL})
\]

Arguments

- \(x\): a vector of numeric or ordered values
- \(\text{na.rm}\): ignored

Value

the minimum value of the vector

Author(s)

Matt Owen <mowen@iq.harvard.edu>

mix

Produce All Combinations of a Set of Lists

Description

Produce All Combinations of a Set of Lists

Usage

\[
\text{mix}(\ldots)
\]

Arguments

- \(\ldots\): a set of lists to mix together

Value

all the combinations of the lists with repetition

Note

This function is used internally by the `mi` constructors in order to produce the complete set of combinations of data-frames and factors by to subset the data-frames.
**Mode**

*Compute the Statistical Mode of a Vector*

**Description**

Compute the Statistical Mode of a Vector

**Usage**

`Mode(x)`

**Arguments**

- `x` a vector of numeric, factor, or ordered values

**Value**

the statistical mode of the vector. If two modes exist, one is randomly selected (by design)

**Author(s)**

Matt Owen <mowen@iq.harvard.edu>

---

**model.frame.multiple**

*Create Model Frame from multiple Object*

**Description**

This method creates a `model.frame` from a `multiple` object. This method will be deprecated as the development of Zelig 4 progresses.

**Usage**

```r
### S3 method for class 'multiple'
model.frame(formula, data, eqn=NULL,...)
```

**Arguments**

- `formula` an object of both type formula and multiple
- `data` a `data.frame`
- `eqn` the number of equations in the formula
- `...` ignored parameters
model.matrix.multiple

Value

a model.frame object

Author(s)

Kosuke Imai, Olivia Lau, Gary King and Ferdinand Alimadhi

model.matrix.multiple  Create Design Matrix of a multiple Object

Description

This method is used to generate a model.matrix adhering to the specifications in the help document "model.matrix".

Usage

```r
## S3 method for class 'multiple'
model.matrix(object, data, shape="compact", eqn=NULL,...)
```

Arguments

- `object`: an object of type `multiple`. This represents a Zelig 3.5 formula
- `data`: a data.frame
- `shape`: a character-string specifying the shape of the matrix
- `eqn`: an integer specifying the number of equations
- `...`: ignored parameters

Note

This method is scheduled to be deprecated.

model.matrix.parseFormula

Construct Design Matrix from a Parsed, Zelig-style Formula

Description

This method constructs a design matrix from a Zelig-style formula. This matrix is commonly used in statistical simulation, and will likely be relevant as the relevant form of a setx object.
Usage

```r
## S3 method for class 'parseFormula'
model.matrix(object, data = NULL, ...
```

Arguments

- `object` : a "parseFormula" object
- `data` : a "data.frame"
- `...` : ignored parameters

Value

A `model.matrix` specifying information relevant to a statistical model

Note

This method is primarily used by the `setx` function.

Author(s)

Matt Owen <mowen@iq.harvard.edu>

Description

This function currently has no documentation, but is essential in Zelig 3.5's implementation of formulae.

Usage

```
multilevel(tt, data, mode, eqn, ...)
```

Arguments

- `tt` : a terms object
- `data` : a `data.frame`
- `mode` : ???
- `eqn` : an integer specifying the number of equations in a model
- `...` : ignored parameters

Value

A list with the "terms" attribute specified
**name.object**

**Author(s)**

Kosuke Imai, Olivia Lau, Gary King and Ferdinand Alimadhi

---

**Name Elements of an Object**

**Description**

Returns an object

**Usage**

`name.object(obj, names)`

**Arguments**

- `obj` a vector or matrix
- `names` a character-vector specifying names

**Value**

the original object, with a "colnames" or "names" equal to the parameter "names". If "names" is larger than "obj", the "names" parameter is truncated appropriately. If it is smaller, then the latter part of "obj" is replaced with a numbered generic column name

**Note**

This method is used internally by Zelig to name the columns and elements of matrices and vectors for simulations and bootstrapped parameters.

**Author(s)**

Matt Owen `<mowen@iq.harvard.edu>`
names.qi  The Names of a 'qi' Object

Description

Function to get the names of a 'qi' object. This function does not entirely parallel the functionality of traditional 'names' methods; this is because the $ operator has been overloaded to support a unique style of value extraction. For technical details, please see the source code.

Usage

```r
## S3 method for class 'qi'
names(x)
```

Arguments

- `x`  a 'qi' object

Value

a character-vector containing the names of the Quantities of Interest

Note

No method exists to set the names of a 'qi' object, once it is constructed. This will be a feature added later.

Author(s)

Matt Owen <mowen@iq.harvard.edu>

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

```r
data(newpainters)
```
Format

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

Source


References


param

Generic Method for Simulating Ancillary/Auxillary Parameters of Zelig Models

Description

The param method is used by developers to specify simulated and fixed ancillary parameters of the Zelig statistical model. That is, this method is used between the zelig2 function and the qi as a helper function that specifies all the necessary details needed to simulate quantities of interest, given the fitted statistical model produced by the zelig2 function.

Usage

param(obj, num, ...)

Arguments

obj a zelig object
num an integer specifying the number of simulations to sample
... optional parameters which will likely be ignored

Value

The main purpose of the param function is to return a list of key-value pairs, specifying information that should be shared between the qi function and the fitted statistical model (produced by the zelig2 function). This list can contain the following entries:

simulations specifies a set of simulated parameters used to describe the statistical model’s underlying distribution
alpha specifies the fixed (non-simulated) ancillary parameters used by the statistical model’s underlying distribution.

family specifies a family object used to implicitly define the link and linkinv functions. That is, this specifies the "link" and "inverse link" functions of generalized linear models.

link specifies the link function to be used. This parameter is largely unimportant compared to the "inverse link" function.

linkinv specifies the linkinv function to be used.

Note

The 'param' function is a method meant to be overloaded by Zelig Developers.

Author(s)

Matt Owen <mowen@iq.harvard.edu>

Examples

```
param.some.model <- function (obj, num, ...) {
  list(
    simulations = NULL,
    alpha = NULL,
    link = NULL,
    linkinv = NULL,
    fam = NULL
  )
}
```

Description

If no param function is set for a Zelig model, then this function will return NULL.

Usage

```
## Default S3 method:
param(obj, num, ...)
```

Arguments

- obj: ignored parameter
- num: ignored parameter
- ...: ignored parameters
**Description**

Param Method for the exp Zelig Model

**Usage**

```r
## S3 method for class 'exp'
param(obj, num, ...)
```

**Arguments**

- `obj`: a `zelig` object
- `num`: an integer specifying the number of simulations to sample
- `...`: ignored parameters

**Value**

a list to be cast as a `parameters` object

**Note**

This method is used by the `param` Zelig model

**Author(s)**

Matt Owen <mowen@iq.harvard.edu>

---

**Description**

Return parameter estimates for the “gamma” GLM in Zelig.

**Usage**

```r
## S3 method for class 'gamma'
param(obj, num, ...)
```
**Arguments**

- **obj**: a `zelig` object
- **num**: an integer specifying the number of simulations to sample
- **...**: ignored parameters

**Value**

a list to be cast as a ‘parameters’ object

**Author(s)**

Matt Owen <mowen@iq.harvard.edu>

---

**Description**

Param Method for the logit Zelig Model

**Usage**

```r
## S3 method for class 'logit'
param(obj, num, ...)
```

**Arguments**

- **obj**: a `zelig` object
- **num**: an integer specifying the number of simulations to sample
- **...**: ignored parameters

**Value**

a list to be cast as a ‘parameters’ object

**Note**

This method is used by the logit Zelig model

**Author(s)**

Matt Owen <mowen@iq.harvard.edu>
### param.ls

**Param Method for the 'ls' Zelig Model**

#### Description

Param Method for the 'ls' Zelig Model

#### Usage

```r
## S3 method for class 'ls'
param(obj, num, ...)
```

#### Arguments

- `obj`: a 'zelig' object
- `num`: an integer specifying the number of simulations to sample
- `...`: ignored parameters

#### Value

a list to be cast as a 'parameters' object

#### Note

This method currently returns via a deprecated style

#### Author(s)

Matt Owen <mowen@iq.harvard.edu>

### param.negbinom

**Param Method for the 'negbinom' Zelig Model**

#### Description

Param Method for the 'negbinom' Zelig Model

#### Usage

```r
## S3 method for class 'negbinom'
param(obj, num=1000, ...)
```
Param Method for the 'normal' Zelig Model

Arguments

obj    a 'zelig' object
num    an integer specifying the number of simulations to sample
...    ignored

Value

a list to be cast as a 'parameters' object

Note

This method is used by the 'negbinom' Zelig model

Author(s)

Matt Owen <mowen@iq.harvard.edu>

---

Param Method for the 'normal' Zelig Model

Description

Param Method for the 'normal' Zelig Model

Usage

```r
## S3 method for class 'normal'
param(obj, num=1000, ...)
```

Arguments

obj    a 'zelig' object
num    an integer specifying the number of simulations to sample
...    ignored

Value

a list to be cast as a 'parameters' object

Note

This method is used by the 'normal' Zelig model

Author(s)

Matt Owen <mowen@iq.harvard.edu>
**Description**

Param Method for the 'poisson' Zelig Model

**Usage**

```r
## S3 method for class 'poisson'
param(obj, num=1000, ...)
```

**Arguments**

- `obj` a 'zelig' object
- `num` an integer specifying the number of simulations to sample
- `...` ignored

**Value**

a list to be cast as a 'parameters' object

**Note**

This method is used by the 'poisson' Zelig model

**Author(s)**

Matt Owen <mowen@iq.harvard.edu>

---

**Description**

Param Method for the 'probit' Zelig Model

**Usage**

```r
## S3 method for class 'probit'
param(obj, num=1000, ...)
```
param.relogit

Arguments

obj a 'zelig' object
num an integer specifying the number of simulations to sample
... ignored

Value

a list to be cast as a 'parameters' object

Note

This method is used by the 'probit' Zelig model

Author(s)

Matt Owen <mowen@iq.harvard.edu>

param.relogit Estimate Parameters for the “relogit” Zelig Model

Description

Returns estimates on parameters, as well as, specifying link and inverse-link functions.

Usage

## S3 method for class 'relogit'
param(obj, num, ...)

Arguments

obj a zelig object containing the fitted model
num an integer specifying the number of simulations to compute
... unspecified parameters

Value

a list specifying important parameters for the “relogit” model

Note

This method merely calls “param.logit”.
param.relogit  

Estimate Parameters for the “relogit” Zelig Model

Description

Returns estimates on parameters, as well as, specifying link and inverse-link functions.

Usage

## S3 method for class 'relogit2'
param(obj, num, x, ...)

Arguments

- **obj**: a zelig object containing the fitted model
- **num**: an integer specifying the number of simulations to compute
- **x**: ideally we should be able to remove this parameter
- **...**: unspecified parameters

Value

a list specifying important parameters for the “relogit” model

param.tobit  

Param Method for the tobit Zelig Model

Description

Param Method for the tobit Zelig Model

Usage

## S3 method for class 'tobit'
param(obj, num, ...)

Arguments

- **obj**: a 'zelig' object
- **num**: an integer specifying the number of simulations to sample
- **...**: ignored parameters

Value

a list to be cast as a 'parameters' object
parameters

Note

This method is used by the tobit Zelig model

Author(s)

Matt Owen <mowen@iq.harvard.edu>

parameters Constructor for ‘parameters’ class

Description

Constructor for ‘parameters’ class

Usage

parameters(simulations, alpha, fam = NULL, link = NULL,
  linkinv = NULL)

Arguments

  simulations a vector or matrix containing simulated values
  alpha ancillary parameters for the Zelig statistical model
  fam a family object which implicitly specifies the link and link-inverse functions for
  the
  link the link function of the specified statistical model. The ‘linkinv’ parameter is
  implicitly defined by by the ‘link’ parameter, when ‘linkinv’ is omitted
  linkinv the inverse link function

Value

a ‘parameters’ object

Author(s)

Matt Owen <mowen@iq.harvard.edu>
**parse.formula**  
*Parse Formulas for Zelig Models*

**Description**

Parse Formulas for Zelig Models

**Usage**

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

**Arguments**

- `formula` a formula
- `model` a Zelig model
- `data` a data-frame

**Note**

This is used typically in multinomial and multivariate Zelig models

**Author(s)**

Kosuke Imai and Olivia Lau

---

**parseFormula**  
*Parse Zelig-style Formulae*

**Description**

Zelig uses three distinct types of formulae. This method is a re-design of the Zelig function `parse.formula`.

**Usage**

```r
parseFormula(obj, data = NULL)
```

**Arguments**

- `obj` a list or formula
- `data` the data set associated with the formula object

**Value**

an object of type "parseFormula". This object has slots specifying:
**parseFormula.list**

**Parse List-Style Zelig Formulae**

**Description**

This method parses a list-style Zelig formula.

**Usage**

```r
## S3 method for class 'list'
parseFormula(obj, data=NULL)
```

**Arguments**

- `obj`  
  a list of formulae
- `data`  
  a data frame

**Value**

an object of type “parseFormula”

**Author(s)**

Matt Owen <mowen@iq.harvard.edu>

---

**parseFormula.Formula**  

**Parse “Formula”-style Zelig Formulae**

**Description**

This method parses a “Formula”-style Zelig formula. This is to support the “Formula” object. It seems like it has the right idea when it comes to expressing multiple responses.

**Usage**

```r
## S3 method for class 'Formula'
parseFormula(obj, data=NULL)
```

**Arguments**

- `obj`  
  a list of formulae
- `data`  
  a data frame

**Value**

an object of type “parseFormula”

**Author(s)**

Matt Owen <mowen@iq.harvard.edu>
**Value**

An object of type "parseFormula"

**Author(s)**

Matt Owen <mowen@iq.harvard.edu>

---

**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).
plot.ci

**Method for plotting pooled simulations by confidence intervals**

**Description**

Plot confidence intervals of simulated quantities of interest, across a range of a variable.

**Usage**

```r
## S3 method for class 'ci'
plot(x, qi="ev", var=NULL, ..., main = NULL, sub = NULL, xlab = NULL,
ylab = NULL, xlim = NULL, ylim = NULL, legcol="gray20", col=NULL, leg=1, legpos=NULL,
                 ci=c(80,95,99.9))
```

**Arguments**

- `x` A `sim` object
- `qi` a character-string specifying the quantity of interest to plot: "ev" expected values, "pv" predicted values, "fd" first differences
- `var` The variable to be used on the x-axis
- `...` Parameters to be passed to the `truehist` function which is implicitly called for numeric simulations
- `main` A character-string, specifying the main title of the plot
- `sub` A character-string, specifying the sub-title of the plot
- `xlab` A character-string, specifying the label for the x-axis
- `ylab` A character-string, specifying the label for the y-axis
- `xlim` A vector of length 2, specifying the left-most and right-most values for the plot
- `ylim` A vector of length 2, specifying the bottom-most and top-most values for the plot
- `legcol` "legend color", an valid color used for plotting the line colors in the legend
col  a valid vector of colors of at least length 3 to use to color the confidence intervals
leg  "legend position", an integer from 1 to 4, specifying the position of the legend.
    1 to 4 correspond to “SE”, “SW”, “NW”, and “NE” respectively
legpos  "legend type", exact coordinates and sizes for legend. Overrides argument “leg.type”
    ci  A vector of length up to 3, specifying the three confidence interval levels to plot
    on the graph (where confidence is expressed on the scale 0-100).

Value
the current graphical parameters. This is subject to change in future implementations of Zelig

Author(s)
James Honaker, adapted by Matt Owen <mowen@iq.harvard.edu>

---

plot.MI.sim  

Plot graphs of simulated multiply-imputed data

Description
This function is currently unimplemented, and reserved for future use.

Usage
### S3 method for class 'MI.sim'
plot(x, ...)

Arguments
x  A zelig ‘sim’ object, with multiply imputed data
...
        ignored parameters

Value
NULL (invisibly)

Author(s)
Matt Owen <mowen@iq.harvard.edu>
Method for plotting pooled simulations by confidence intervals

Description
Plot pooled simulated quantities of interest.

Usage
```r
## S3 method for class 'pooled.sim'
plot(x, qi="ev", var=NULL, ..., main = NULL, sub = NULL,
    xlab = NULL, ylab = NULL, xlim = NULL, ylim = NULL, legcol="gray20", col=NULL, leg=1,
    legpos=NULL, ci=c(80,95,99.9))
```

Arguments
- `x` A ‘sim’ object
- `qi` a character-string specifying the quantity of interest to plot
- `var` The variable to be used on the x-axis. Default is the variable across all the
  chosen values with smallest nonzero variance
- `...` Parameters to be passed to the ‘truehist’ function which is implicitly called for
  numeric simulations
- `main` A character-string, specifying the main title of the plot
- `sub` A character-string, specifying the sub-title of the plot
- `xlab` A character-string, specifying the label for the x-axis
- `ylab` A character-string, specifying the label for the y-axis
- `xlim` A vector of length 2, specifying the left-most and right-most values for the plot
- `ylim` A vector of length 2, specifying the bottom-most and top-most values for the plot
- `legcol` “legend color”, an valid color used for plotting the line colors in the legend
- `col` a valid vector of colors of at least length 3 to use to color the confidence intervals
- `leg` “legend position”, an integer from 1 to 4, specifying the position of the legend. 1 to 4 correspond to “SE”, “SW”, “NW”, and “NE” respectively
- `legpos` “legend type”, exact coordinates and sizes for legend. Overrides argument “leg.type”
- `ci` A numeric triple, specifying the three levels to plot confidence intervals for, scaled 0 to 100

Value
the current graphical parameters. This is subject to change in future implementations of Zelig

Author(s)
James Honaker, adapted by Matt Owen <mowen@iq.harvard.edu>
plot.sim  Method for plotting simulations

Description
Plot simulated quantities of interest.

Usage

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

Arguments

x a ‘sim’ object
...

parameters to be passed to the ‘truehist’ function which is implicitly called for numeric simulations

Value
nothing

Author(s)
Matt Owen <mowen@iq.harvard.edu>

plot.simulations  Plot Any Simulation from the Zelig Core Package

Description
Plots any simulation from the core package. In general, this function can neatly plot simulations containing five of the popular “quantities of interest” - “Expected Values: E(Y|X)”, “Predicted Values: Y|X”, “Expected Values (for X1): E(Y|X1)”, “Predicted Values (for X1): Y|X1” and “First Differences: E(Y|X1) - E(Y|X)”.

Usage
plot.simulations(x, ...)

Arguments

x an object
...

parameters passed to the “plot” and “barplot” functions
**Value**

the original graphical parameters

**Author(s)**

Matt Owen <mowen@iq.harvard.edu>

---

**print.qi**  \hspace{1cm} \textit{Print a Quantity of Interest in Human-Readable Form}

**Description**

Print simulated quantities of interest in a human-readable form

**Usage**

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

**Arguments**

- `x`  \hspace{1cm} a `qi` object
- `...`  \hspace{1cm} ignored parameters

**Value**

the object that was printed (invisibly)

**Author(s)**

Matt Owen <mowen@iq.harvard.edu>

---

**print.qi.summarized**  \hspace{1cm} \textit{Print Method for Summarized Quantities of Interest}

**Description**

Print Method for Summarized Quantities of Interest

**Usage**

```r
## S3 method for class 'qi.summarized'
print(x, ...)
```
print.setx

Arguments

  x        a 'summarized.qi' object
  ...      parameters to be passed to the specific print functions

Value

  x (invisibly)

Author(s)

  Matt Owen <mowen@iq.harvard.edu>

See Also

  special_print_MATRIX and special_print_LIST
print.setx.mi  \textit{Print a Bundle of Data-sets}

\textbf{Description}

Print a Bundle of Data-sets

\textbf{Usage}

\begin{verbatim}
## S3 method for class 'setx.mi'
print(x, ...)
\end{verbatim}

\textbf{Arguments}

\begin{itemize}
  \item \textit{x} \hspace{1cm} a \textit{setx} object to print
  \item \textit{...} \hspace{1cm} ignored parameters
\end{itemize}

\textbf{Value}

the \textit{setx} object (invisibly)

\textbf{Author(s)}

Matt Owen \textless{}mowen@iq.harvard.edu\textgreater{}

\ \ \ \print.sim \hspace{1cm} \textit{Print values of 'sim' objects}

\textbf{Description}

This function is currently unimplemented, and included for future development

\textbf{Usage}

\begin{verbatim}
## S3 method for class 'sim'
print(x, ...)
\end{verbatim}

\textbf{Arguments}

\begin{itemize}
  \item \textit{x} \hspace{1cm} a \textit{sim} object (ignored)
  \item \textit{...} \hspace{1cm} ignored parameters
\end{itemize}

\textbf{Value}

NULL (invisibly)
print.summary.MCMCZelig

Print a Summary MCMCZelig Object

Description

This method prints a summary object for MCMCZelig objects.

Usage

```r
## S3 method for class 'summary.MCMCZelig'
print(x, digits=max(3, getOption("digits") - 3), ...)
```

Arguments

- `x`: an "MCMCZelig" object
- `digits`: a numeric specifying the precision of the summary object
- `...`: ignored parameters

Value

- a summary.MCMCZelig object

print.summary.pooled.sim

Print a Summary of a Set of Pooled Simulated Interests

Description

Prints the summary information from a set of pooled simulated interests. This method assumes that quantities of interest are kept in a data type which can be used with “rbind”.

Usage

```r
## S3 method for class 'summary.pooled.sim'
print(x, ...)
```

Arguments

- `x`: a “summary.pooled.sim” object, containing summarized information about simulated quantities of interest
- `...`: Optional parameters that will be passed onward to “print.matrix” (the matrix printing function)
Value
a "summary.pooled.sim" object storing the quantities of interest

Author(s)
Matt Owen <mowen@iq.harvard.edu>

print.summary.relogit
Print Summary of a Rare-event Logistic Model

Description
Prints the

Usage
## S3 method for class 'summary.relogit'
print(x, digits = max(3,
            getOption("digits") - 3), ...)

Arguments
x an "relogit.summary" object produced by the "summary" method.
digits an integer specifying the number of digits of precision to specify
... parameters passed forward to the "print.glm" function

Value
x (invisibly)

print.summary.sim

Arguments

\( x \)  
the object to print

\( \text{digits} \)  
an integer specifying the number of digits of precision

\( \ldots \)  
ignored parameters

Value

\( x \) (invisibly)

Description

Print values of simulated quantities of interest (stored in a “summary.sim” object.

Usage

```r
## S3 method for class 'summary.sim'
print(x, ...)
```

Arguments

\( x \)  
a 'summary.sim' object

\( \ldots \)  
ignored parameters

Value

the value of the ‘summary.sim’ object (invisibly)

Author(s)

Matt Owen <mowen@iq.harvard.edu>
print.summarySim.MI  \hspace{1cm} \textit{Print Multiply Imputed Simulations Summary}

\textbf{Description}

Prints summary information about Multiply Imputed Fits

\textbf{Usage}

\begin{verbatim}
### S3 method for class 'summarySim.MI'
print(x, digits=3, ...)
\end{verbatim}

\textbf{Arguments}

\begin{itemize}
\item \texttt{x} \hspace{1cm} a 'summarySim.MI' object
\item \texttt{digits} \hspace{1cm} an integer specifying the number of digits of precision to print
\item ... \hspace{1cm} ignored parameters
\end{itemize}

\textbf{Author(s)}

Matt Owen <mowen@iq.harvard.edu>

---

print.zelig  \hspace{1cm} \textit{Print values of “zelig” objects}

\textbf{Description}

Print the zelig object as a list

\textbf{Usage}

\begin{verbatim}
### S3 method for class ‘zelig’
print(x, ...)  \hspace{1cm} \texttt{\ldots}
\end{verbatim}

\textbf{Arguments}

\begin{itemize}
\item \texttt{x} \hspace{1cm} a ‘zelig’ object
\item ... \hspace{1cm} ignored parameters
\end{itemize}

\textbf{Value}

the ‘zelig’ object (invisibly)

\textbf{Author(s)}

Matt Owen <mowen@iq.harvard.edu>
qi  

Generic Method for Computing Quantities of Interest

Description
The qi function is used by developers to simulated quantities of interest. This method, as a result, is the most significant method of any Zelig statistical model.

Usage
qi(obj, x = NULL, x1 = NULL, y = NULL, num, param = NULL)

Arguments
- obj: a zelig object
- x: a setx object or NULL
- x1: an optional setx object
- y: this parameter is reserved for simulating average treatment effects, though this feature is currently supported by only a handful of models
- num: an integer specifying the number of simulations to compute
- param: a parameters object

Value
a list of key-value pairs specifying pairing titles of quantities of interest with their simulations

Note
Run example(qi) to see a trivial version of

Author(s)
Matt Owen <mowen@iq.harvard.edu>

Examples
qi.some.model <- function(obj, x=NULL, x1=NULL, y=NULL, param=NULL) {
  list(
    "Expected Values: E(Y|X)" = NA,
    "Predicted Values: Y|X" = NA
  )
}
qi.exp

*Compute quantities of interest for 'exp' Zelig models*

**Description**

Compute quantities of interest for 'exp' Zelig models

**Usage**

```r
## S3 method for class 'exp'
qi(obj, x=NULL, x1=NULL, y=NULL, num=1000,
    param=NULL)
```

**Arguments**

- `obj`: a `zelig` object
- `x`: a `setx` object or NULL
- `x1`: an optional `setx` object
- `y`: this parameter is reserved for simulating average treatment effects, though this feature is currently supported by only a handful of models
- `num`: an integer specifying the number of simulations to compute
- `param`: a parameters object

**Value**

A list of key-value pairs specifying pairing titles of quantities of interest with their simulations

**Author(s)**

Matt Owen <mowen@iq.harvard.edu>

qi.summarize

*Constructor for QI Summarized Class This class takes an arbitrary number of the _same_ type of quantities of interest labels them, then merges them into one simple printable block. In particular, this class determines which print function to use based on the the type and size od data to be passed to the print function.*

**Description**

Constructor for QI Summarized Class This class takes an arbitrary number of the _same_ type of quantities of interest labels them, then merges them into one simple printable block. In particular, this class determines which print function to use based on the type and size of data to be passed to the print function.
reduceMI

Usage

qi.summarize(title, x, ...)

Arguments

title a character-string specifying the title of the QI
x a list of summarized quantities of interest
... additional quantities of interest (the parameter that titles these will be used as the name of the data.frame

Value

the list of QI's (invisibly)

Author(s)

Matt Owen <mowen@iq.harvard.edu>

reduceMI

Reduce MI Formulas Take a formula in any of the reduced form or in a structural form and return the most reduced form of that formula

Description

Reduce MI Formulas Take a formula in any of the reduced form or in a structural form and return the most reduced form of that formula

Usage

reduceMI(f)

Arguments

f a formula

Note

This formula is used primarily by 'zelig2' functions of multivariate Zelig models

Author(s)

Ferdinand Alimadhi, Kosuke Imai, and Olivia Lau
relogit

*Fit a rare-event logistic model in Zelig*

**Description**

Fits a rare-event (”relogit”) model.

**Usage**

```r
relogit(formula, data = sys.parent(), tau = NULL,
    bias.correct = TRUE, case.control = "prior", ...)
```

**Arguments**

- `formula`: a formula object
- `data`: ...
- `tau`: ...
- `bias.correct`: ...
- `case.control`: ...
- `...`: ???

**Value**

A “relogit” “glm” object

---

repl

*Generic Method for Replicating Data*

**Description**

Generic Method for Replicating Data

**Usage**

```r
repl(object, ...)
```

**Arguments**

- `object`: a ’zelig’ object
- `...`: parameters

**Value**

A replicated object
repl.default  Default Method for Replicating Statistics

Description

Replicate a simulation

Usage

```r
## Default S3 method:
repl(object, data=NULL, ...)
```

Arguments

- object: an object to replicate
- data: a data.frame
- ...: ignored parameters

Value

a replicated object

Author(s)

Kosuke Imai and Olivia Lau <mowen@iq.harvard.edu>

---

repl.sim  Method for Replicating Simulated Quantities of Interest

Description

Replicate simulated quantities of interest

Usage

```r
## S3 method for class 'sim'
repl(object, x=NULL, x1=NULL, y=NULL,
     num=1000, prev = NULL, bootstrap = FALSE, boot.fn=NULL,
     cond.data = NULL, ...)
```
Arguments

- `object` a 'zelig' object
- `x` a 'setx' object
- `x1` a secondary 'setx' object used to perform particular computations of quantities of interest
- `y` a parameter reserved for the computation of particular quantities of interest (average treatment effects). Few models currently support this parameter
- `num` an integer specifying the number of simulations to compute
- `prev` ignored
- `bootstrap` ignored
- `boot.fn` ignored
- `cond.data` ignored
- `...` special parameters which are reserved for future versions of Zelig

Value

a 'sim' object storing the replicated quantities of interest

Author(s)

Matt Owen <mowen@iq.harvard.edu>

---

replace.call

Hook to Update the Zelig Call with the Appropriate Call Object

Description

Hook to Update the Zelig Call with the Appropriate Call Object

Usage

`replace.call(zobj, call1, call2)`

Arguments

- `zobj` a 'zelig' object
- `call1` the original call to Zelig
- `call2` the manuafactured call to the model fitting function

Value

the 'zelig' object with a modified 'call' slot

Note

This function is used internally by Zelig, and currently deprecated.
**robust.gee.hook**  
*Classify Fitted Object as Naive or Robust*

**Description**

This hook is ran after the call to the external mode. It sets the class of the object (in addition to its other designations) as 'gee.naive' or 'gee.robust' depending on the value of the robust parameter.

**Usage**

```r
robust.gee.hook(obj, Zall, Call, robust, ...)
```

**Arguments**

- `obj`  
  a zelig object
- `Zall`  
  the call made to the zelig function
- `Call`  
  the call made to the external model
- `robust`  
  a logical specifying whether to use the naive or robust covariance matrix
- `...`  
  ignored parameters

**Value**

a zelig object with the additional class gee.robust or gee.naive

**Author(s)**

Skyler

---

**robust(glm).hook**  
*Hook for “glm” Models in Zelig*

**Description**

Adds support for robust error-estimates in the Zelig “glm” models.

**Usage**

```r
robust(glm).hook(obj, zcall, call, robust = FALSE, ...)
```

**Arguments**

- `obj`  
  a zelig object
- `zcall`  
  the original call to the zelig model
- `call`  
  the call that will be evaluated for the
- `robust`  
  a logical specifying whether or not to use robust error estimates
- `...`  
  ignored parameters
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="ROC Curve", xlab="Proportion of 1's Correctly Predicted", ylab="Proportion of 0's Correctly Predicted", plot = TRUE, ... )

Arguments

y1 response variable for the first model
y2 response variable for the second model
fitted1 fitted values for the first model. These values may represent either the in-sample or out-of-sample fitted values
fitted2 fitted values for the second model
cutoff 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
lty1 the line type of the first model (defaults to 'line’)
lty2 the line type of the second model (defaults to 'dashed’)
lwd1 the line width of the first model (defaults to 1)
lwd2 the line width of the second model (defaults to 1)
col1 the color of the first model (defaults to 'black’)
col2 the color of the second model (defaults to 'black’)
main a title for the plot (defaults to "ROC Curve")
xlab a label for the X-axis
ylab a label for the Y-axis
plot whether to generate a plot to the selected device
... additional parameters to be passed to the plot
Value

if plot is TRUE, rocplot simply generates a plot. Otherwise, a list with the following is produced:

- **roc1**: 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.

- **roc2**: 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.

- **area1**: the area under the first ROC curve, calculated using Reimann sums.

- **area2**: the area under the second ROC curve, calculated using Reimann sums.

---

**Sanction**

*Multilateral Economic Sanctions*

**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**

```r
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**

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

setx(obj, fn = NULL, data = NULL, cond = FALSE, ...)

Arguments

obj the saved output from zelig
fn a list of functions to apply to the data frame
data a new data frame used to set the values of explanatory variables. If data = NULL (the default), the data frame called in zelig is used
cond a logical value indicating whether unconditional (default) or conditional (choose cond = TRUE) prediction should be performed. If you choose cond = TRUE, setx will coerce fn = NULL and ignore the additional arguments in .... If cond = TRUE and data = NULL, setx will prompt you for a data frame.
... user-defined values of specific variables for overwriting the default values set by the function fn. For example, adding var1 = mean(data$var1) or x1 = 12 explicitly sets the value of x1 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]])

Author(s)

Matt Owen <mowen@iq.harvard.edu>, Olivia Lau and Kosuke Imai

See Also

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

```r
# 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)
```

Description

Set explanatory variables

Usage

```r
## Default S3 method:
setx(obj, fn=NULL, data=NULL,
     cond=NULL, ...)
```

Arguments

- `obj` a `zelig` object
- `fn` a list of key-value pairs specifying which function apply to columns of the keys
- `data` a data.frame
- `cond` ignored
- `...` parameters specifying what to explicitly set each column as. This is used to produce counterfactuals

Value

a `setx` object

Author(s)

Matt Owen <mowen@iq.harvard.edu>, Kosuke Imai, and Olivia Lau
**setx.MI**

Set Explanatory Variables for Multiply Imputed Data-sets
This function simply calls `setx.default` once for every fitted model within the 'zelig.MI' object.

### Description
Set Explanatory Variables for Multiply Imputed Data-sets
This function simply calls `setx.default` once for every fitted model within the 'zelig.MI' object.

### Usage
```r
## S3 method for class 'MI'
setx(obj, ..., data=NULL)
```

### Arguments
- **obj**: a 'zelig' object
- **...**: user-defined values of specific variables for overwriting the default values set by the function `fn`
- **data**: a new data-frame

### Value
a 'setx.mi' object used for computing Quantities of Interest by the 'sim' method.

### Author(s)
Matt Owen &lt;mowen@iq.harvard.edu&gt;

### See Also
- `setx`

---

**sim**

Generic Method for Computing and Organizing Simulated Quantities of Interest
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.
Description

Generic Method for Computing and Organizing Simulated Quantities of Interest

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

```r
sim(obj, x = NULL, x1 = NULL, y = NULL, num = 1000,
    bootstrap = F, bootfn = NULL, cond.data = NULL, ...)
```

Arguments

- `obj` the output object from zelig
- `x` values of explanatory variables used for simulation, generated by setx
- `x1` optional values of explanatory variables (generated by a second call of setx) particular computations of quantities of interest
- `y` a parameter reserved for the computation of particular quantities of interest (average treatment effects). Few models currently support this parameter
- `num` an integer specifying the number of simulations to compute
- `bootstrap` currently unsupported
- `bootfn` currently unsupported
- `cond.data` currently unsupported
- `...` arguments reserved future versions of Zelig

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 setx values for the explanatory variables, used to calculate the quantities of interest (expected values, predicted values, etc.).
- `x1` the optional setx object used to simulate first differences, and other model-specific quantities of interest, such as risk-ratios.
- `call` the options selected for sim, used to replicate quantities of interest.
- `zelig.call` the original command and options for zelig, 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 x.
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 \( x \) and \( x_1 \). The difference is calculated by subtracting the expected values given \( x \) from the expected values given \( x_1 \). (If do not specify \( x_1 \), 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

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

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 \( Y_i(t_i = 0) \), 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 \( t_i = 0 \).

Author(s)

Matt Owen <mowen@iq.harvard.edu>, Olivia Lau and Kosuke Imai

Method for Simulating Quantities of Interest from 'zelig' Objects

Description

Simulate quantities of interest
Usage

```r
## Default S3 method:
sim(obj, x=NULL, x1=NULL, y=NULL,
    num=1000, bootstrap = FALSE, bootfn=NULL, cond.data =
    NULL, ...)```

Arguments

- `obj` a 'zelig' object
- `x` a 'setx' object
- `x1` a secondary 'setx' object used to perform particular computations of quantities of interest
- `y` a parameter reserved for the computation of particular quantities of interest (average treatment effects). Few models currently support this parameter
- `num` an integer specifying the number of simulations to compute
- `bootstrap` ignored
- `bootfn` ignored
- `cond.data` ignored
- `...` parameters to be passed to the boot function, if one is supplied

Value

a 'sim' object storing the replicated quantities of interest

Author(s)

Matt Owen <mowen@iq.harvard.edu>

---

`sim.MI`  
*Simulate Multiply Imputed Data*

Description

Simulate Multiply Imputed Data

Usage

```r
## S3 method for class 'MI'
sim(obj, x=NULL, x1=NULL, y=NULL, num=1000,
    ...)```
simulation.matrix

Arguments

 obj: a `zelig.MI` object containing several fits for two or more sub-setted data-frames
 x: a `setx.mi` object containing explanatory variables for each fitted model
 x1: a `setx.mi` object containing explanatory variables for each fitted model
 y: this feature is currently unimplemented
 num: an integer specifying the number of simulations to compute
 ...: ignored parameters

Value

a `sim.MI` with simulated quantities of interest for each fitted contained by `obj`

Author(s)

Matt Owen <mowen@iq.harvard.edu>

See Also

sim

---

**simulation.matrix**

*Get Simulations as a Matrix*

Description

Returns a MxN matrix where N is the number of simulations and M is the number of predicted values. Additionally, a “labels” attribute is attached that produces a human-readable identifier for each column.

Usage

`simulation.matrix(obj, which = NULL, ...)`

Arguments

 obj: an object, typically a “sim” or “pooled.sim” object.
 which: a character-vector specifying the titles of quantities of interest to extract
 ...: additional parameters

Value

a simulation matrix

Author(s)

Matt Owen <mowen@iq.harvard.edu>
Return Simulations of Parameter Coefficients

Description

Returns simulated parameters of coefficients for use in statistical simulation. The values are set by the model-fitting function and the developer of the qi.<model name> method.

Usage

```r
# S3 method for class 'parameters'
simulations(object, ...)
```

Arguments

- `object`: a `parameters` object
- `...`: ignored

Value

Simulations, specified by the Zelig model, of the ancillary parameters

Note

This function does not differ at all from coef.default

Author(s)

Matt Owen <mowen@iq.harvard.edu>

Plot Quantities of Interest in a Zelig-fashion

Description

Various graph generation for different common types of simulated results from Zelig

Usage

```r
simulations.plot(y, y1=NULL, xlab='', ylab='', main='',
                 col=NULL, line.col=NULL, axisnames=TRUE)
```
Arguments

y A matrix or vector of simulated results generated by Zelig, to be graphed.
y1 For comparison of two sets of simulated results at different choices of covariates, this should be an object of the same type and dimension as y. If no comparison is to be made, this should be NULL.
xlab Label for the x-axis.
yle 
lab Label for the y-axis.
main Main plot title.
col A vector of colors. Colors will be used in turn as the graph is built for main plot objects. For nominal/categorical data, this colors renders as the bar color, while for numeric data it renders as the background color.
line.col A vector of colors. Colors will be used in turn as the graph is built for line color shading of plot objects.
axisnames a character-vector, specifying the names of the axes

Value

nothing

Author(s)

James Honaker

Description

This data set contains five sociomatrixes 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
special_print_LIST

Method for Printing Summarized QI's in a List Form

Description
Method for Printing Summarized QI's in a List Form

Usage

`.print.qi.summarized.LIST(x, ...)`

Arguments

x a `summarized.qi` object
...

additional parameters to be used by the `print.matrix` method

Value

x (invisibly)

Note
This function is used internally by Zelig

Author(s)
Matt Owen <mowen@iq.harvard.edu>

---

special_print_MATRIX

Method for Printing Summarized QI's in a Matrix Form

Description
Method for Printing Summarized QI's in a Matrix Form

Usage

`.print.qi.summarized.MATRIX(x, ...)`

Arguments

x a `summarized.qi` object
...

additional parameters
splitUp

Value

x (invisibly)

Note

This function is used internally by Zelig

Author(s)

Matt Owen <mowen@iq.harvard.edu>

---

splitUp

Split a List into Two Lists

This function takes any list, and splits into two lists - one containing the values of arguments with specifically specified values and those without specified values.

Usage

splitUp(args)

Arguments

`args` a list

Value

A list containing two entries: the key-value paired entries (titled wordful) and the unkeyed entries (titled wordless)

Note

This function is a good candidate for deprecation

Author(s)

Matt Owen <mowen@iq.harvard.edu>

Examples

#list(wordful = list(x=1, y=2), wordless=list(2, "red"))
Description

This function takes the value of an object and stores it within a specified environment. This is similar to simply using the `assign` function, but will not overwrite existing values in the specified environment. It accomplishes this by appending a prefix to the name of the variable until the name becomes unique.

Usage

```r
store.object(obj, envir, name = NULL, prefix = ".")
```

Arguments

- `obj` any object
- `envir` an environment object, which will contain the object with the assigned name
- `name` a character-string specifying the name that the object will be stored as in the specified environment
- `prefix` a character string specifying the prefixes to append to names that already have matches in the destination environment

Value

a character-string specifying the name of the object in the destination environment

Note

This method does not correct invalid names. That is, there is no test to determine whether the submitted name is valid.

Author(s)

Matt Owen <mowen@iq.harvard.edu>
**structuralToReduced**  
*Transform the Multilevel’s Structural Formulas Into Reduced Form*

**Description**
Transform the Multilevel’s Structural Formulas Into Reduced Form

**Usage**
```r
structuralToReduced(f)
```

**Arguments**
- `f`: a list of formulas

**Value**
a formula in reduced form

**Author(s)**
Ferdinand Alimadhi, Kosuke Imai, and Olivia Lau

---

**summarize**  
*Generic method for summarizing simulated quantities of interest*

**Description**
Generic method for summarizing simulated quantities of interest

**Usage**
```r
summarize(obj)
```

**Arguments**
- `obj`: a qi object, storing simulations of quantities of interest

**Value**
a summarized.qi object

**Author(s)**
Matt Owen <mowen@iq.harvard.edu>
**summarize.default**

*Summarize Simulated Quantities of Interest*

**Description**

Summarize Simulated Quantities of Interest

**Usage**

```r
## Default S3 method:
summarize(obj)
```

**Arguments**

- `obj` a qi object, storing simulations of quantities of interest

**Value**

a 'summarized.qi' object

**Author(s)**

Matt Owen <mowen@iq.harvard.edu>

---

**summary.glm.robust**

*Summary of Generalized Linear Model with Robust Error Estimates*

**Description**

Returns summary of a glm model with robust error estimates. This only slightly differs from how the standard GLM’s behave.

**Usage**

```r
## S3 method for class 'glm.robust'
summary(object, ...)
```

**Arguments**

- `object` a “glm.robust” fitted model
- `...` parameters to pass to the standard “summary.glm” method

**Value**
a object of type “summary.glm.robust” and “summary.glm”
summary.MI

Summary of Multiply Imputed Statistical Models

Description

...

Usage

## S3 method for class 'MI'
summary(object, subset = NULL, ...)

Arguments

object a set of fitted statistical models
subset an integer vector, specifying the indices of the data.frames to be used in the subset
... parameters to forward

Value

a list of summaries

Author(s)

Matt Owen <mowen@iq.harvard.edu>

summary.MI.sim

Method for summarizing simulations of multiply imputed quantities of interest

Description

Method for summarizing simulations of multiply imputed quantities of interest

Usage

## S3 method for class 'MI.sim'
summary(object, ...)

Arguments

object a ‘MI.sim’ object
... ignored parameters
summary.pooled.sim

Value

a ‘summarized.MI.sim’ object

Author(s)

Matt Owen <mowen@iq.harvard.edu>

summary.pooled.sim  Return a Summary of a Set of Pooled Simulated Interests

Description

Returns the summary information from a set of pooled simulated interests. The object returned contains the slots “labels”, a character-vector specifying the labels (explanatory variable titles) of the qi’s, “titles”, a character vector specifying the names of the quantities of interest, and

Usage

## S3 method for class ‘pooled.sim’
summary(object, ...)

Arguments

object      a “pooled.sim” object, containing information about simulated quantities of interest

...         Ignored parameters

Value

a “summary.pooled.sim” object storing the replicated quantities of interest

Author(s)

Matt Owen <mowen@iq.harvard.edu>
Summary for "Relogit" Fitted Model

**Description**

Summarize important components of the “relogit” model

**Usage**

```r
## S3 method for class 'Relogit'
summary(object, ...)
```

**Arguments**

- `object`: a “Relogit” object
- `...`: other parameters

**Value**

a “summary.relogit” object

---

Summary for "Relogit2" Fitted Model

**Description**

Summarize important components of the “relogit” model

**Usage**

```r
## S3 method for class 'Relogit2'
summary(object, ...)
```

**Arguments**

- `object`: a “Relogit2” object
- `...`: other parameters

**Value**

a “summary.relogit2” object
**summary.sim**

*Method for summarizing simulations of quantities of interest*

**Description**

Return a “summary.sim” object (typically for display)

**Usage**

```r
## S3 method for class 'sim'
summary(object, ...)
```

**Arguments**

- `object`: a ‘MI.sim’ object
- `...`: ignored parameters

**Value**

a ’summarized.MI.sim’ object

**Author(s)**

Matt Owen <mowen@iq.harvard.edu>

---

**summary.zelig**

*Zelig Object Summaries*

**Description**

Compute summary data for zelig objects

**Usage**

```r
## S3 method for class 'zelig'
summary(object, ...)
```

**Arguments**

- `object`: a zelig object
- `...`: parameters forwarded to the generic summary object

**Value**

the summary of the fitted model
**Author(s)**
Matt Owen <mowen@iq.harvard.edu>

---

**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**
```r
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**

---

**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**
```r
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

References


---

t.setx

Matrix Transpose of a “setx” Object

Description

Returns a “setx” object as column vector. If multiple values for each explanatory term has been set, then return a NxM matrix where ‘N’ is the number of explanatory terms and ‘M’ is the number of values set for each term.

Usage

```r
## S3 method for class 'setx'
t(x)
```

Arguments

- `x` a `setx` object

Value

- a transposed matrix

Author(s)

Matt Owen <mowen@iq.harvard.edu>
### table.levels

Create a table, but ensure that the correct columns exist. In particular, this allows for entries with zero as a value, which is not the default for standard tables.

**Description**

Create a table, but ensure that the correct columns exist. In particular, this allows for entries with zero as a value, which is not the default for standard tables.

**Usage**

`table.levels(x, levels, ...)`

**Arguments**

- `x`: a vector
- `levels`: a vector of levels
- `...`: parameters for table

**Value**

a table

**Author(s)**

Matt Owen <mowen@iq.harvard.edu>

---

### terms.multiple

Extract terms from a multiple Object.

**Description**

Extracts terms from Zelig-3.5-style formulae. This function is scheduled for removal.

**Usage**

```r
## S3 method for class 'multiple'
terms(x, data=NULL, ...)
```

**Arguments**

- `x`: a Zelig v3.5 formula
- `data`: a data.frame
- `...`: ignored parameters
**Author(s)**

Kosuke Imai, Olivia Lau, Gary King and Ferdinand Alimadhi

---

### terms.vglm

**Model Terms for 'vglm' Models**

**Description**

Model Terms for 'vglm' Models

**Usage**

```r
## S3 method for class 'vglm'
terms(x, ...)
```

**Arguments**

- `x`: a fitted model object from the VGAM library
- `...`: ignored parameters

**Value**

the models terms of this fitted model object

---

**Author(s)**

Ferdinand Alimadhi, Kosuke Imai and Olivia Lau

---

### terms.zelig

**Model Terms for a Zelig Object**

**Description**

This method simply extracts the model terms for the fitted model passed to the `zelig` function.

**Usage**

```r
## S3 method for class 'zelig'
terms(x, ...)
```

**Arguments**

- `x`: a `zelig` object
- `...`: forwarded parameters

**Value**

terms of the original fitted model
termsFromFormula  

*Extract Terms from Zelig-style Formulae*

---

**Description**

This method is a sugary function to extract terms from any type of Zelig-style formula.

**Usage**

```r
termsFromFormula(obj)
```

**Arguments**

- `obj`  
  a Zelig-style formula

---

**Author(s)**

Matt Owen <mowen@iq.harvard.edu>

---

**TexCite  
*Get a TeX-style Citation*

---

**Description**

Get a TeX-style Citation

**Usage**

```r
TexCite(model)
```

**Arguments**

- `model`  
  a character-string specifying the name of the Zelig model of which to describe in TeX-style

**Value**

a string to be rendered as part of a LaTeX-style document
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**
```r
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**

tobuildformula

**Build Formula ???**

**Description**
This function builds a formula

**Usage**
```r
tobuildformula(Xnames, sepp = "+")
```

**Arguments**
- **Xnames**: a character-vector
- **sepp**: a separator (???)

**Value**
a character-string

**Author(s)**
???
tolmerformat  

Convert a Formula into 'lmer' Representation from Reduced Form
Take a formula in its reduced form and return it as a 'lmer' representation (from the lme4 package). This is basically removing the starting 'tag' from each term.

Description

Convert a Formula into 'lmer' Representation from Reduced Form. Take a formula in its reduced form and return it as a 'lmer' representation (from the lme4 package). This is basically removing the starting 'tag' from each term.

Usage

tolmerformat(f)

Arguments

f  a formula in reduced form

Value

the 'lmer' representation of 'f'

Author(s)

Ferdinand Alimadhi, Kosuke Imai, and Olivia Lau

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


---

**ucfirst**

*Uppercase First Letter of a String*

**Description**

This method sets the first character of a string to its uppercase, sets all other characters to lowercase.

**Usage**

```
ucfirst(str)
```

**Arguments**

- `str` a vector of character-strings

**Value**

a vector of character strings

**Author(s)**

Matt Owen <mowen@iq.harvard.edu>

---

**user.prompt**

*Prompt User*

**Description**

Prompts user to hit enter

**Usage**

```
user.prompt(msg = NULL)
```

**Arguments**

- `msg` a character-string, specifying a message to be displayed
Value

This function is used for its side effects

Note

This function is primarily used by Zelig demo scripts

---

**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

```r
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](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

z Return value for a zelig2-function

Description
This is an API-function that bridges a model-fitting function with a zelig interface.

Usage
z(.function, ..., .hook = NULL)
**Arguments**

- `.function` a function
- `...` a set of parameters to be evaluated symbolically
- `.hook` a function to be applied after the external, model-fitting function is called

**Value**

a “z” object which specifies how to evaluate the fitted model

**Note**

This is used internally by Zelig-dependent packages to instruct Zelig how to evaluate the function call to a particular statistical model.

---

**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**

```
zelig(formula, model, data, ..., by = NULL, cite = T)
```

**Arguments**

- `formula` a symbolic representation of the model to be estimated, in the form `y ~ \, x1 + x2`, where `y` is the dependent variable and `x1` and `x2` are the explanatory variables, and `y`, `x1`, and `x2` 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 `x1*x2` without computing them in prior steps; `I(x1*x2)` to include only the interaction term and exclude the main effects; and quadratic terms in the form `I(x1^2)`
- `model` the name of a statistical model. Type `help.zelig("models")` to see a list of currently supported models
- `data` 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 `mi`)
- `...` additional arguments passed to `zelig`, depending on the model to be estimated
zelig.call

by a factor variable contained in \textit{data}. Zelig will subset the data frame based on the levels in the \textit{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:
\begin{verbatim}
z.out <- zelig(y ~ x1 + x2, data = mydata, model = "ls", by = "state")
\end{verbatim}
You may also use \texttt{by} to run models using MatchIt subclass.

\textbf{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 \texttt{summary(z.out)} or individually extracted using, for example, \texttt{z.out$coefficients}. See the specific models listed above for additional output values, or simply type \texttt{names(z.out)}.

\textbf{Author(s)}

Matt Owen (<mowen@iq.harvard.edu>), Kosuke Imai, Olivia Lau, and Gary King

Maintainer: Matt Owen (<mowen@iq.harvard.edu>)

---

\textbf{Description}

Create Function Call

\textbf{Usage}

\begin{verbatim}
zelig.call(Call, zelig2, remove = NULL)
\end{verbatim}

\textbf{Arguments}

\begin{itemize}
  \item \texttt{Call} a call object, typically specifying the original function call to zelig
  \item \texttt{zelig2} the return-value of the zelig2 method
  \item \texttt{remove} a list of character vectors specifying which parameters to ignore from the original call to zelig
\end{itemize}

\textbf{Value}

a function call used to fit the statistical model

\textbf{Author(s)}

Matt Owen (<mowen@iq.harvard.edu>)
Description

'zelig.skeleton' generates the necessary files used to create a Zelig package. Based on (and using) R's 'package.skeleton' it removes some of the monotony of building statistical packages. In particular, 'zelig.skeleton' produces templates for the zelig2, describe, param, and qi methods. For more information about creating these files on an individual basis, please refer to the tech manuals, which are available by typing: ?zelig2, ?param, or ?qi.

Usage

```
zeligNskeletonHpkgL models ] cHIL
author ] Bunknown authorBL path ] BNBL force ] falseL
email ] Bmaintainer@softwareMprojectNorgBL
depends ] cL, ..., .gitignore ] TRUE,
.Rbuildignore ] TRUE)
```

Arguments

- **pkg**: a character-string specifying the name of the Zelig package
- **models**: a vector of strings specifying models to be included in the package
- **author**: a vector of strings specifying contributors to the package
- **path**: a character-string specifying the path to the package
- **force**: a logical specifying whether to overwrite files and create necessary directories
- **email**: a string specifying the email address of the package's maintainer
- **depends**: a vector of strings specifying package dependencies
- **...**: ignored parameters
- **.gitignore**: a logical specifying whether to include a copy of a simple .gitignore in the appropriate folders (inst/doc and the package root)
- **.Rbuildignore**: a logical specifying whether to include a copy of a simple .Rbuildignore in the appropriate folders (inst/doc and the package root)

Value

nothing

Author(s)

Matt Owen <mowen@iq.harvard.edu>
Table of links for Zelig

Description

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

zelig2

Interface Between Zelig Models and External Functions

Description

The zelig2 function acts as a simple interface between a user’s call to the zelig function and the zelig functions subsequent call to the pre-existing external model. The external model varies based on which model is being called.

Value

The main purpose of the zelig2 function is to return a list of key-value pairs, specifying how Zelig should interface with the external method. This list has the following format:

- .function specifies the name of the external method to be called by zelig function. Subsequent parameters are called and evaluated as a function call to the function of the named string.
- .hook specifies the name of a hook function as a string. The hook function is only evaluated on zelig object once the external method fits the statistical model.
- ... any parameters aside from .function and .hook is as part of the function call to the external model.

Note

Writing zelig2 functions is required of Zelig developers. In particular, zelig2 functions act as an interface between external models (models not included in the Zelig package) and the zelig function which must use that model.

zelig2 is not an actual function. Rather,

Examples

zelig2some.model <- function (formula, weights, verbose, ..., data) {
  list(
    .function = 'some.other.method',
    .hook = NULL,
    formula = formula,
    weights = 2 * weights,
    data = data
  )
## Description

Interface between Zelig and the bayesian models.

## Usage

```r
zelig2factor.bayes(formula, factors = 2, burnin = 1000, mcmc = 20000, verbose = 0, ..., data)
zelig2logit.bayes(formula, burnin = 1000, mcmc = 10000, verbose = 0, ..., data)
zelig2mlogit.bayes(formula, burnin = 1000, mcmc = 10000, verbose = 0, ..., data)
zelig2normal.bayes(formula, burnin = 1000, mcmc = 10000, verbose = 0, ..., data)
zelig2oprobit.bayes(formula, burnin = 1000, mcmc = 10000, verbose = 0, ..., data)
zelig2poisson.bayes(formula, burnin = 1000, mcmc = 10000, verbose = 0, ..., data)
```

## Arguments

- `formula`: a formula
- `factors`: additional parameters
- `data`: a data.frame
- `burnin`: a parameter corresponding to the 'burnin' parameter for the MCMCprobit function
- `mcmc`: a parameter corresponding to the 'mcmc' parameter for the MCMCprobit function
- `verbose`: a parameter corresponding to the 'verbose' parameter for the MCMCprobit function

## Value

A list specifying '.function'
Description

Interface between Zelig and Basic GLM Fitting Functions

Usage

zelig2exp(formula, ..., robust = FALSE, cluster = NULL, data)
zelig2gamma(formula, ..., data)
zelig2logit(formula, weights = NULL, robust = F, ..., data)
zelig2lognorm(formula, ..., robust = FALSE, cluster = NULL, data)
zelig2ls(formula, ..., data, weights = NULL)
zelig2negbinom(formula, weights = NULL, ..., data)
zelig2normal(formula, weights = NULL, ..., data)
zelig2poisson(formula, weights = NULL, ..., data)
zelig2probit(formula, weights = NULL, ..., data)
zelig2relogit(formula, ..., tau = NULL, bias.correct = NULL, case.control = NULL, data)
zelig2tobit(formula, ..., below = 0, above = Inf, robust = FALSE, cluster = NULL, data)
zelig2twosls(formula, ..., data)

Arguments

formula a formula
... additonal parameters
weights a numeric vector
robust a boolean specifying whether to use robust error estimates
cluster a vector describing the clustering of the data
data a data.frame
tau ... 
bias.correct ...
case.control ...
below a numeric or infinite specifying a lower boundary for censored responses
above a numeric or infinite specifying an upper boundary for censored responses

Value

a list used by Zelig to call the model-fitting function
zelig2-gee

Bridge between Zelig and the GEE Model Fitting Functions

Description

Bridge between Zelig and the GEE Model Fitting Functions

Usage

zelig2gamma.gee(formula, id, robust, ..., R, corstr = "independence", data)
zelig2logit.gee(formula, id, robust, ..., R, corstr = "independence", data)
zelig2normal.gee(formula, id, robust, ..., R, corstr = "independence", data)
zelig2poisson.gee(formula, id, robust, ..., R, corstr = "independence", data)
zelig2probit.gee(formula, id, robust, ..., R, corstr = "independence", data)

Arguments

formula a formula
id a character-string specifying the column of the data-set to use for clustering
robust a logical specifying whether to robustly or naively compute the covariance matrix. This parameter is ignored in the zelig2 method, and instead used in the robust.hook function, which executes after the call to the gee function
... ignored parameters
R a square-matrix specifying the correlation
corstr a character-string specifying the correlation structure
data a data.frame

Value

a list specifying the call to the external model

zelig2-survey

Interface between Zelig and svyglm

Description

Interface between zelig and svyglm for the logit.survey
Usage

zelig2.gamma.survey(formula, weights = NULL, ids = NULL, 
probs = NULL, strata = NULL, fpc = NULL, nest = FALSE, 
check.strata = !nest, repweights = NULL, type, 
combined.weights = FALSE, rho = NULL, 
bootstrap.average = NULL, scale = NULL, rscales = NULL, 
fpctype = "fraction", return.replicates = FALSE, 
na.action = "na.omit", start = NULL, etastart = NULL, 
mustart = NULL, offset = NULL, model1 = TRUE, 
method = "glm.fit", x = FALSE, y = TRUE, 
contrasts = NULL, design = NULL, link = "inverse", 
data, ...)

zelig2.logit.survey(formula, weights = NULL, ids = NULL, 
probs = NULL, strata = NULL, fpc = NULL, nest = FALSE, 
check.strata = !nest, repweights = NULL, type, 
combined.weights = FALSE, rho = NULL, 
bootstrap.average = NULL, scale = NULL, rscales = NULL, 
fpctype = "fraction", return.replicates = FALSE, 
na.action = "na.omit", start = NULL, etastart = NULL, 
mustart = NULL, offset = NULL, model1 = TRUE, 
method = "glm.fit", x = FALSE, y = TRUE, 
contrasts = NULL, design = NULL, data)

zelig2.normal.survey(formula, weights = NULL, ids = NULL, 
probs = NULL, strata = NULL, fpc = NULL, nest = FALSE, 
check.strata = !nest, repweights = NULL, type, 
combined.weights = FALSE, rho = NULL, 
bootstrap.average = NULL, scale = NULL, rscales = NULL, 
fpctype = "fraction", return.replicates = FALSE, 
na.action = "na.omit", start = NULL, etastart = NULL, 
mustart = NULL, offset = NULL, model1 = TRUE, 
method = "glm.fit", x = FALSE, y = TRUE, 
contrasts = NULL, design = NULL, data)

zelig2.poisson.survey(formula, weights = NULL, ids = NULL, 
probs = NULL, strata = NULL, fpc = NULL, nest = FALSE, 
check.strata = !nest, repweights = NULL, type, 
combined.weights = FALSE, rho = NULL, 
bootstrap.average = NULL, scale = NULL, rscales = NULL, 
fpctype = "fraction", return.replicates = FALSE, 
na.action = "na.omit", start = NULL, etastart = NULL, 
mustart = NULL, offset = NULL, model1 = TRUE, 
method = "glm.fit", x = FALSE, y = TRUE, 
contrasts = NULL, design = NULL, data)

zelig2.probit.survey(formula, weights = NULL, ids = NULL, 
probs = NULL, strata = NULL, fpc = NULL, nest = FALSE,
check.strata = !nest, repweights = NULL, type, combined.weights = FALSE, rho = NULL, bootstrap.average = NULL, scale = NULL, rscales = NULL, fpctype = "fraction", return.replicates = FALSE, na.action = "na.omit", start = NULL, etastart = NULL, mustart = NULL, offset = NULL, model1 = TRUE, method = "glm.fit", x = FALSE, y = TRUE, contrasts = NULL, design = NULL, data)

Arguments

formula a formula
weights ...
ids ...
probs ...
strata ...
fpc ...
nest ...
check.strata ...
repweights ...
type ...
combined.weights ...
rho ...
bootstrap.average ...
scale ...
rscales ...
fpctype ...
return.replicates ...
na.action ...
start ...
etastart ...
mustart ...
offset ...
model1 ...
method ...
x ...
y ...
contrasts ...
zeligBuildWeights

**Description**

The zeligBuildWeights utility allows developers building models or modules for Zelig, to easily define what types of weights can be set by the user and passed to estimation function. In some cases it can reconfigure the dataset by duplication to functionally invoke discrete weighting by replication of observations, when the estimator itself can not utilize weights.

**Usage**

zeligBuildWeights(weights=NULL, repweights=NULL, zeros="zeros", rebuild=FALSE, allowweights=TRUE, allowrepweights=TRUE, data=NULL)

**Arguments**

- **weights**: A set of non-negative value weights. Overrides repweights if defined.
- **repweights**: A set of whole number (non-negative integer) weights. Useful if weights are just for making copies of or deleting certain observations or for frequency weights.
- **zeros**: An option on how to deal with zero valued user supplied weights. Default of "zero" allows zero weights, "epsilon" changes zeroes to 1e-08, "remove" removes those observations from the dataset.
- **rebuild**: An option to allow specified repweights to reconfigure the rows of the dataset to rebuild a corresponding dataset where every row is of weight 1. Useful if analysis model does not accept weights.
- **allowweights**: Defines if weights are allowed in model.
- **allowrepweights**: Defines if repweights are allowed in model. Overridden if useweights=TRUE.
- **data**: Dataset, required if weights are defined by variable name, or if dataset is to be reconfigured (by rebuild or zeros options).

**Value**

A list used to construct parameters for the svyglm function.

**Note**

This manual file is largely incomplete, and needs a significant amount of filling out. This, in itself, might be motivation to divide this package into more models with more specific function.
Value

- weights: A vector of weights of the structure defined by the developer and required by the analysis model. Or NULL if certain checks are failed.
- data: A reconfigured dataset, if modified.

Author(s)

James Honaker <jhonaker@iq.harvard.edu>

See Also

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

---

### ZeligDescribeModel

**Produce a 'description' Object from the Name of a Model**

**Description**

Produce a 'description' Object from the Name of a Model

**Usage**

ZeligDescribeModel(model.name)

**Arguments**

- model.name: a character-string specifying a Zelig model

**Value**

A 'description' object specified by the 'model.name' parameter. This object is created by executing the specified Zelig models’ 'describe' function

**Note**

The 'description' object is a list-style object containing citation information
**ZeligListModels**

*Get a Character-Vector of All Models with a 'zelig2' Function*

---

**Description**

Get a Character-Vector of All Models with a 'zelig2' Function

**Usage**

```r
ZeligListModels(zelig.only = FALSE)
```

**Arguments**

- `zelig.only`: a boolean specifying whether we want to search only the Zelig namespace

**Value**

A character-vector of the Zelig models loaded on the user’s machine

**Note**

In order for a Zelig model to either execute correctly or be listed as a legal Zelig model, the function name must be prefixed with 'zelig2'.

**Author(s)**

Matt Owen <mowen@iq.harvard.edu>

---

**ZeligListTitles**

*List the Titles of the Zelig Statistical Models*

---

**Description**

List the Titles of the Zelig Statistical Models

**Usage**

```r
ZeligListTitles()
```

**Value**

A list of manual titles for the Zelig software
Index

*Topic **datasets**
- approval, 8
- bivariate, 22
- coalition, 28
- coalition2, 29
- eidat, 40
- free1, 41
- free2, 43
- friendship, 45
- grunfeld, 51
- hoff, 52
- homerun, 53
- immigration, 54
- klein, 58
- kmenta, 58
- macro, 62
- MatchIt.url, 64
- mexico, 67
- mid, 68
- newpainters, 74
- PErisk, 87
- sanction, 107
- sna.ex, 116
- SupremeCourt, 126
- swiss, 126
- tobin, 131
- turnout, 132
- voteincome, 134
- Weimar, 135
- Zelig.url, 139

*Topic **file**
- setx, 108

*Topic **package**
- zelig, 136
- Zelig-package, 6

*Topic **weights**
- zeligBuildWeights, 145

*print.qi.summarized.MATRIX
  (special_print_MATRIX), 117

  alpha, 7
  approval, 8
  as.bootlist, 8
  as.bootvector, 9
  as.data.frame.setx, 9
  as.description, 10
  as.description.description, 11
  as.description.list, 11
  as.matrix.pooled.setx, 12
  as.matrix.setx, 13
  as.parameters, 13
  as.parameters.default, 14
  as.parameters.list, 15
  as.parameters.parameters, 16
  as.qi, 16
  as.qi.default, 17
  as.qi.list, 18
  as.qi.qi, 18
  as.summarized, 19
  as.summarized.list, 20
  as.summarized.summarized.qi, 20
  attach.env, 21

  bivariate, 22
  bootfn.default, 22
  bootstrap, 23
  bootstrap.default, 24
  bootstrap.gamma, 24
  bootstrap.negbinom, 25
  bootstrap.normal, 25

  callToString, 26
  cite, 27
  cluster.formula, 27
  cmvglm, 28
  coalition, 28
  coalition2, 29
INDEX

coef.parameters, 30
combine, 30
constructDataFrame, 31
constructDesignMatrix, 32

depends.on.zelig, 32
describe, 33
describe.default, 33
describe.exp, 34
describe.gamma, 34
describe.logit, 35
describe.ls, 35
describe.negbinom, 36
describe.normal, 37
describe.poisson, 37
describe.probit, 38
describe.tobit, 38
describe.zelig, 39
description, 39
eidat, 40

find.match, 41
free1, 41
free2, 43
friendship, 45
get.package, 45
GetObject, 46
getPredictorTerms, 46
getResponse.Formula
     (getResponseTerms.Formula), 47
getResponseTerms, 47
getResponseTerms.Formula, 47
getResponseTerms.formula, 48
getResponseTerms.list, 49
GetSlot, 49
GetSlot.zelig, 50
grunfeld, 51

has.zelig2, 51
help.zelig, 52
hoff, 52
homerun, 53

ignore, 54
immi1(immigration), 54
immi2(immigration), 54
immi3(immigration), 54
immi4(immigration), 54

immii5(immigration), 54
immigration, 54
is.formula, 55
is.qi, 55
is.valid.qi.list, 56
is.zelig.compliant, 56
is.zelig.package, 57

klein, 58
kmenta, 58

link, 59
linkinv, 59
list.depth, 60
list.zelig.dependent.packages, 60
list.zelig.models, 61
load.dependencies(loadDependencies), 61
loadDependencies, 61

macro, 62
make.parameters, 62
makeModelMatrix, 63
makeZeligObject, 63
MatchIt.url, 64
Max, 64
MCMChook, 65
McmChookFactor, 65
Median, 66
mexico, 67
mi, 67
mid, 68
Min, 69
mix, 69
Mode, 70
model.frame.multiple, 70
model.matrix.multiple, 71
model.matrix.parseFormula, 71
multilevel, 72

name.object, 73
names.qi, 74
newpainters, 74

param, 75
param.default, 76
param.exp, 77
param.gamma, 77
param.logit, 78
param.ls, 79
param.negbinom, 79
param.normal, 80
param.poisson, 81
param.probit, 81
param.relogit, 82
param.relogit2, 83
param.tobit, 83
parameters, 84
parse.formula, 85
parseFormula, 85
parseFormula.Formula, 86
parseFormula.list, 86
PERisk, 87
plot.ci, 88
plot.MI.sim, 89
plot.pooled.sim, 90
plot.sim, 91
plot.simulations, 91
print.qi, 92
print.qi.summarized, 92
print.setx, 93
print.setx.mi, 94
print.sim, 94
print.summary.MCMCZelig, 95
print.summary.pooled.sim, 95
print.summary.relogit, 96
print.summary.relogit2, 96
print.summary.sim, 97
print.summarySim.MI, 98
print.zelig, 98
qi, 75, 99
qi.exp, 100
qi.summarize, 100
reduceMI, 101
relogit, 102
repl, 102
repl.default, 103
repl.sim, 103
replace.call, 104
robust.gee.hook, 105
robust.glm.hook, 105
rocplot, 106
sanction, 107
setx, 108, 110, 111
setx.default, 109
setx.MI, 110
sim, 110, 111, 114
sim.default, 112
sim.MI, 113
simulation.matrix, 114
simulations.parameters, 115
simulations.plot, 115
sna.ex, 116
special_print_LIST, 93, 117
special_print_MATRIX, 93, 117
splitUp, 118
store.object, 119
structuralToReduced, 120
summarize, 120
summarize.default, 121
summary.glm.robust, 121
summary.MI, 122
summary.MI.sim, 122
summary.pooled.sim, 123
summary.Relogit, 124
summary.Relogit2, 124
summary.sim, 125
summary.zelig, 125
SupremeCourt, 126
swiss, 126
t.setx, 127
table.levels, 128
terms.multiple, 128
terms.vglm, 129
terms.zelig, 129
termsFromFormula, 130
TeXCite, 130
tobin, 131
toBuildFormula, 131
tolmerFormat, 132
turnout, 132
ucfirst, 133
user.prompt, 133
voteincome, 134
Weimar, 135
z, 135
Zelig (Zelig-package), 6
zelig, 108, 111, 136
Zelig-package, 6
zelig.call, 137
zelig.skeleton, 138
Zelig.url, 139
zelig2, 139
zelig2-bayes, 140
zelig2-core, 141
zelig2-gee, 142
zelig2-survey, 142
zelig2exp (zelig2-core), 141
zelig2factor.bayes (zelig2-bayes), 140
zelig2gamma (zelig2-core), 141
zelig2gamma.gee (zelig2-gee), 142
zelig2gamma.survey (zelig2-survey), 142
zelig2logit (zelig2-core), 141
zelig2logit.bayes (zelig2-bayes), 140
zelig2logit.gee (zelig2-gee), 142
zelig2logit.survey (zelig2-survey), 142
zelig2lognorm (zelig2-core), 141
zelig2ls (zelig2-core), 141
zelig2mlogit.bayes (zelig2-bayes), 140
zelig2negbinom (zelig2-core), 141
zelig2normal (zelig2-core), 141
zelig2normal.bayes (zelig2-bayes), 140
zelig2normal.gee (zelig2-gee), 142
zelig2normal.survey (zelig2-survey), 142
zelig2oprobit.bayes (zelig2-bayes), 140
zelig2poisson (zelig2-core), 141
zelig2poisson.bayes (zelig2-bayes), 140
zelig2poisson.gee (zelig2-gee), 142
zelig2poisson.survey (zelig2-survey), 142
zelig2probit (zelig2-core), 141
zelig2probit.bayes (zelig2-bayes), 140
zelig2probit.gee (zelig2-gee), 142
zelig2probit.survey (zelig2-survey), 142
zelig2relogit (zelig2-core), 141
zelig2tobit (zelig2-core), 141
zelig2twosls (zelig2-core), 141
zeligBuildWeights, 145
ZeligDescribeModel, 146
ZeligListModel, 147
ZeligListTitles, 147