Package ‘obsSens’

February 20, 2015

Type Package
Title Sensitivity analysis for Observational studies.
Version 1.3
Date 2012-01-18
Author Greg Snow <greg.snow@imail.org>
Maintainer Greg Snow <greg.snow@imail.org>
Description Observational studies are limited in that there could be an unmeasured variable related to both the response variable and the primary predictor. If this unmeasured variable were included in the analysis it would change the relationship (possibly changing the conclusions). Sensitivity analysis is a way to see how much of a relationship needs to exist with the unmeasured variable before the conclusions change. This package provides tools for doing a sensitivity analysis for regression (linear, logistic, and cox) style models.
License GPL-2
Repository CRAN
Repository/R-Forge/Project obsSens
Repository/R-Forge/Revision 14
Repository/R-Forge/DateTimeStamp 2013-01-16 22:29:25
Date/Publication 2013-01-18 18:19:47
NeedsCompilation no

R topics documented:

obsSens-package .................................................. 2
obsSensCCC ....................................................... 3
print.obsSens .................................................... 6

Index 8
Perform sensitivity analysis on observational studies to explore possible effects of unmeasured (lurking) variables.

Description

This package provides functions for doing sensitivity analysis on coefficients of regression type models (regression, logistic regression, cox proportional hazards). These assume a true model of the form: \( g(y) = \beta x + \gamma u + \theta z \) where \( u \) is an unmeasured potential lurking variable, \( x \) is the main variable of interest (treatment) and \( z \) represents other potential variables in the model. The response variable (\( y \)) can be continuous, binary, or a survival object. These functions examine the effect of \( u \) on \( \beta \) for different values of \( \gamma \) and the relationship between \( u \) and \( y \).

Details

Package: obsSens
Type: Package
Version: 1.0
Date: 2007-12-21
License: Artistic-2.0

The key functions are all of the form obsSensYXU where \( Y \) specifies the type of variable used as the response variable (\( y \)), \( X \) specifies the type of variable used as the main predictor variable to be tested (\( x \)), and \( U \) specifies the type of unmeasured variable to use. They can take on the following values: S - survival analysis (\( Y \) only), C - Categorical (logistic regression, currently only handles 2 levels), or N - normal (or continuous variables).

All the functions take either a fitted model object (lm, glm, or coxph) or a coefficient value and its confidence interval. You then specify values (vector) for the possible relationship between \( Y \) and \( U \) and \( X \) and \( U \). The return value is a list with a matrix or array with the adjusted coefficients and upper and lower confidence limits.

Author(s)

Greg Snow <greg.snow@imail.org>

References


Examples

# Recreate tables from above references

```r
obsSensCCC( log(23.1), log(c(6.9, 77.7)), g0=c(2,6,10),
            p0=seq(0,.5,.1), p1=seq(0,1,.2) )
```

```r
obsSensSCC( log(1.21), log(c(1.09,1.25)),
            p0=seq(0,.5,.1), p1=seq(0,1,.1), g0=3 )
```

```r
obsSensCNN( log(1.14), log(c(1.10,1.18)),
            rho=c(0,.5,.75,.85,.9,.95,.98,.99),
            gamma=seq(0,1,2), sdx=4.5 )
```

---

**obssensccc**

*Compute Sensitivity analysis for Observational Studies*

**Description**

Computes a Sensitivity analysis for a coefficient from a regression model (linear, logistic, Cox) on Observational data. Computes new coefficient estimate and confidence interval for various relationships between the response, predictor of interest, and an unmeasured variable. The last 3 letters of each function refer to the type of the variables in the order Y, X, U with C meaning categorical, N meaning normal (continuous), and S being a survival object.

**Usage**

```r
obsSensCCC(model, which = 2, g0 = c(2, 6, 10), g1, p0 = seq(0, 1, 0.2), p1 = p0, logOdds = FALSE, method = c("approx", "sim"))
```

```r
obsSensCCN(model, which = 2, gamma = round(seq(0, 2 * bstar, length = 6), 4), delta = seq(0, 3, 0.5), rho = c(0, 0.5, 0.75, 0.85, 0.9, 0.95, 0.98, 0.99), sdx, log = TRUE, method = c("approx", "sim"))
```

```r
obsSensCNN(model, which = 2, gamma = round(seq(0, 2 * bstar, length = 6), 4), rho = c(0, 0.5, 0.75, 0.85, 0.9, 0.95, 0.98, 0.99), sdx, log = TRUE, method = c("approx", "sim"))
```

```r
obsSensNCC(model, which = 2, g0 = c(2, 6, 10), g1, p0 = seq(0, 1, 0.2), p1 = p0, log = TRUE, method = c("approx", "sim"))
```

```r
obsSensNCN(model, which = 2, gamma = round(seq(0, 3, length = 6), 4), delta = seq(0, 3, 0.5), log = TRUE, method = c("approx", "sim"))
```

```r
obsSensNNN(model, which = 2, gamma = round(seq(0, 2 * bstar, length = 6), 4), rho = c(0, 0.5, 0.75, 0.85, 0.9, 0.95, 0.98, 0.99), sdx, log = TRUE, method = c("approx", "sim"))
```

```r
obsSensSCC(model, which = 2, g0 = c(2, 6, 10), g1, p0 = seq(0, 1, 0.2), p1 = p0, logHaz = FALSE, method = c("approx", "sim"))
```

```r
obsSensSCN(model, which = 2, gamma = round(seq(0, 3, length = 6), 4), delta = seq(0, 3, 0.5), logHaz = TRUE, method = c("approx", "sim"))
```

```r
obsSensSNN(model, which = 2, gamma = round(seq(0, 3, length = 6), 4), rho = c(0, 0.5, 0.75, 0.85, 0.9, 0.95, 0.98, 0.99), sdx, log = TRUE, method = c("approx", "sim"))
```
Arguments

model  A regression model object (result of \texttt{lm}, \texttt{glm}, or \texttt{coxph} or other object with a \texttt{coef} method) or a single numerical value representing the estimated coefficient from a regression model.

which  Which coefficient (in the results from \texttt{coef(model)}) corresponds to the predictor of interest. If model is a number then this should be a 2 element vector with the initial confidence interval for model.

g0  The slopes on the unmeasured variable when x=0 (on the log scale, not response).

g1  The slopes on the unmeasured variable when x=1, if missing then g1=g0=gamma is assumed.

p0  Probability that U=1 given x=0.

p1  Probability that U=1 given x=1.

logodds  Should the resulting table be on the log scale or response scale.

log  Should the resulting table be on the log scale or not.

logHaz  Should the resulting table be on the log Hazard ratio scale or Hazard ratio scale.

method  Either "approx" or "sim", only approx is currently implemented.

gamma  Slopes for U (unmeasured variable).

delta  The difference between the mean of U|x=0 and mean of U|x=1.

rho  Correlation coefficient between x and U.

sdx  Standard Deviation of x (default will try to extract this from model, it will need to be provided if model is a number rather than a model object).

Details

These functions are all used to do sensitivity analysis on regression models for observational data. Currently it works with linear regression, logistic regression (using \texttt{glm}), and survival regressions (using \texttt{coxph}). All models are of the general form:

\[ y = b1 \times x + \gamma U + \beta Z \]

Where y is the response (or function of the response). The first letter in the triplet at the end of each function name corresponds to the type of response variable, N-normal/numeric (continuous), C-Categorical (binary, logistic regression), and S-Survival (coxph models).

The x variable is the coefficient of interest (usually the treatment variable, or primary predictor of interest). The 2nd letter in the triplet can be C-Categorical (x is 0 or 1, usually control vs. treatment) or N-Numeric, a continuous variable.

The U variable is an unmeasured potential confounder that we believe may be related to y and x. The final letter in the final triplet refers to the type of unmeasured variable that we want to correct for: C-Categorical/binary (0-1, present-absent) or N-Numerical/continuous (in this case it is assumed to be normal with mean 0 and sd 1).

The Z represents additional covariates in the model that are not of primary interest in the sensitivity analysis. It is assumed that Z and U are independent of each other.

For all the functions you specify the potential relationships between U and y (\texttt{gamma}, g0, and g1) and potential relationships between x and U (p0, p1, delta, and rho). Then the functions compute
a new estimate of b1 (the slope for x, the variable of interest) and its confidence interval given each combination of the relationships between U, y, and x. Currently only the approximation method by Lin et. al. is available. In the future a simulation method will also be implemented.

Value

An obsSens object (S3) stored as a list with the following elements:

- **beta**: A matrix/array with the adjusted slopes for the different conditions.
- **lcl**: A matrix/array with the lower confidence intervals for the slopes.
- **ucl**: A matrix/array with the upper confidence intervals for the slopes.
- **log**: A logical indicating if the values are on the log scale.
- **xname**: The 'name' of the x variable of interest.
- **type**: Character field with the type of y variable, can be 'cat', 'surv', or 'num'.

Note

Note: Currently there are no checks on whether the conditions will be appropriate for the approximation to be close. See the first paper below for the conditions when the approximation is good.

Author(s)

Greg Snow <greg.snow@imail.org>

References


See Also

- print.obsSens
- summary.obsSens

Examples

# Recreate tables from above references

```r
obsSensCCC( log(23.1), log(c(6.9, 77.7)), g0=c(2,6,10), 
p0=seq(0.5,.1), p1=seq(0.1,.2) )

obsSensCC( log(1.21), log(c(1.09,1.25)), 
p0=seq(0.5,.1), p1=seq(0.1,.1), g0=3 )

obsSensCNN( log(1.14), log(c(1.10,1.18)),
```
rho=c(0,.5,.75,.85,.9,.95,.98,.99),
gamma=seq(0,1,2), sdx=4.5 )

print.obsSens

\textit{Print and Summarize ObsSens objects.}

\textbf{Description}

Print a nice table of the results from a sensitivity analysis on observational data.

\textbf{Usage}

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

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

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

\textbf{Arguments}

\begin{itemize}
  \item \textbf{x} \hspace{1cm} An obsSens object or summary.obsSens object.
  \item \textbf{object} \hspace{1cm} An obsSens object.
  \item \textbf{digits} \hspace{1cm} Passed to format.
  \item \textbf{...} \hspace{1cm} Additional arguments passed on to print methods for matrix/array.
\end{itemize}

\textbf{Details}

These functions print and summarize the results of the obsSens functions. The only difference in the print and summary methods is the return value (the information printed to the screen is the same). The \texttt{print} function returns a copy of the original object. The \texttt{summary} function returns a matrix/array of character strings corresponding to what is printed that can hopefully be used along with other functions (latex, xtable, odfTable, ...) for typesetting results in other formats.

\textbf{Value}

The \texttt{print} method returns a copy of the original object. The \texttt{summary} method returns a character array/matrix with the coefficients and confidence intervals to be passed to other typesetting/printing functions.

\textbf{Author(s)}

Greg Snow \texttt{<greg.snow@mail.org>
Examples

# Recreate tables from above references

obsSensCCC( log(23.1), log(c(6.9, 77.7)), g0=c(2,6,10),
            p0=seq(0,.5,.1), p1=seq(0,1,.2) )

obsSensSCC( log(1.21), log(c(1.09,1.25)),
            p0=seq(0,.5,.1), p1=seq(0,1,.1), g0=3 )

obsSensCNN( log(1.14), log(c(1.10,1.18)),
            rho=c(0,.5,.75,.85,.9,.95,.98,.99),
            gamma=seq(0,1,2), sdx=4.5 )
Index

*Topic **models**
   obsSens-package, 2
   obsSensCCC, 3
   print.obsSens, 6

*Topic **package**
   obsSens-package, 2

*Topic **regression**
   obsSens-package, 2
   obsSensCCC, 3

*Topic **survival**
   obsSens-package, 2
   obsSensCCC, 3

obsSens (obsSens-package), 2
obsSens-package, 2
obsSensCCC, 3
obsSensCCN (obsSensCCC), 3
obsSensCNN (obsSensCCC), 3
obsSensNCC (obsSensCCC), 3
obsSensNCN (obsSensCCC), 3
obsSensNNN (obsSensCCC), 3
obsSensSCC (obsSensCCC), 3
obsSensSCN (obsSensCCC), 3
obsSensSNN (obsSensCCC), 3

print.obsSens, 5, 6
print.summary.obsSens (print.obsSens), 6

summary.obsSens, 5
summary.obsSens (print.obsSens), 6