

# Package ‘betaDelta’

June 21, 2024

**Title** Confidence Intervals for Standardized Regression Coefficients

**Version** 1.0.5.9000

**Description** Generates confidence intervals for standardized regression coefficients using delta method standard errors for models fitted by `lm()` as described in Yuan and Chan (2011) <[doi:10.1007/s11336-011-9224-6](https://doi.org/10.1007/s11336-011-9224-6)> and Jones and Waller (2015) <[doi:10.1007/s11336-013-9380-y](https://doi.org/10.1007/s11336-013-9380-y)>. The package can also be used to generate confidence intervals for differences of standardized regression coefficients and as a general approach to performing the delta method. A description of the package and code examples are presented in Pesigan, Sun, and Cheung (2023) <[doi:10.1080/00273171.2023.2201277](https://doi.org/10.1080/00273171.2023.2201277)>.

**URL** <https://github.com/jeksterslab/betaDelta>,  
<https://jeksterslab.github.io/betaDelta/>

**BugReports** <https://github.com/jeksterslab/betaDelta/issues>

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Roxygen** list(markdown = TRUE)

**Depends** R (>= 3.5.0)

**Imports** numDeriv

**Suggests** knitr, rmarkdown, testthat, betaSandwich

**RoxygenNote** 7.3.1

**NeedsCompilation** no

**Author** Ivan Jacob Agaloos Pesigan [aut, cre, cph]  
(<<https://orcid.org/0000-0003-4818-8420>>)

**Maintainer** Ivan Jacob Agaloos Pesigan <[r.jeksterslab@gmail.com](mailto:r.jeksterslab@gmail.com)>

Contents

BetaDelta . . . . .	2
coef.betadelta . . . . .	4
coef.deltamethod . . . . .	4
coef.diffbetadelta . . . . .	5
confint.betadelta . . . . .	6
confint.deltamethod . . . . .	7
confint.diffbetadelta . . . . .	8
Delta . . . . .	9
DeltaGeneric . . . . .	10
DiffBetaDelta . . . . .	12
nas1982 . . . . .	13
print.betadelta . . . . .	13
print.deltamethod . . . . .	14
print.diffbetadelta . . . . .	15
summary.betadelta . . . . .	16
summary.deltamethod . . . . .	17
summary.diffbetadelta . . . . .	18
vcov.betadelta . . . . .	19
vcov.deltamethod . . . . .	19
vcov.diffbetadelta . . . . .	20
<b>Index</b>	<b>22</b>

---

BetaDelta	<i>Estimate Standardized Regression Coefficients and the Corresponding Sampling Covariance Matrix</i>
-----------	-------------------------------------------------------------------------------------------------------

---

Description

Estimate Standardized Regression Coefficients and the Corresponding Sampling Covariance Matrix

Usage

```
BetaDelta(object, type = "mvn", alpha = c(0.05, 0.01, 0.001))
```

Arguments

object	Object of class <code>lm</code> .
type	Character string. If <code>type = "mvn"</code> , use the multivariate normal-theory approach. If <code>type = "adf"</code> , use the asymptotic distribution-free approach.
alpha	Numeric vector. Significance level $\alpha$ .

**Value**

Returns an object of class `betadelta` which is a list with the following elements:

**call** Function call.

**args** Function arguments.

**lm\_process** Processed `lm` object.

**gamma** Asymptotic covariance matrix of the sample covariance matrix.

**acov** Asymptotic covariance matrix of the standardized slopes.

**vcov** Sampling covariance matrix of the standardized slopes.

**est** Vector of standardized slopes.

**Author(s)**

Ivan Jacob Agaloos Pesigan

**References**

Jones, J. A., & Waller, N. G. (2015). The normal-theory and asymptotic distribution-free (ADF) covariance matrix of standardized regression coefficients: Theoretical extensions and finite sample behavior. *Psychometrika*, 80(2), 365–378. doi:[10.1007/s113360139380y](https://doi.org/10.1007/s113360139380y)

Pesigan, I. J. A., Sun, R. W., & Cheung, S. F. (2023). `betaDelta` and `betaSandwich`: Confidence intervals for standardized regression coefficients in R. *Multivariate Behavioral Research*. doi:[10.1080/00273171.2023.2201277](https://doi.org/10.1080/00273171.2023.2201277)

Yuan, K.-H., & Chan, W. (2011). Biases and standard errors of standardized regression coefficients. *Psychometrika*, 76(4), 670–690. doi:[10.1007/s1133601192246](https://doi.org/10.1007/s1133601192246)

**See Also**

Other Beta Delta Functions: [DiffBetaDelta\(\)](#)

**Examples**

```
object <- lm(QUALITY ~ NARTIC + PCTGRT + PCTSUPP, data = nas1982)
std <- BetaDelta(object)
# Methods -----
print(std)
summary(std)
coef(std)
vcov(std)
confint(std, level = 0.95)
```

---

coef.betadelta	<i>Standardized Regression Slopes</i>
----------------	---------------------------------------

---

**Description**

Standardized Regression Slopes

**Usage**

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

**Arguments**

object	Object of class betadelta.
...	additional arguments.

**Value**

Returns a vector of standardized regression slopes.

**Author(s)**

Ivan Jacob Agaloos Pesigan

**Examples**

```
object <- lm(QUALITY ~ NARTIC + PCTGRT + PCTSUPP, data = nas1982)  
std <- BetaDelta(object)  
coef(std)
```

---

coef.deltamethod	<i>Estimates</i>
------------------	------------------

---

**Description**

Estimates

**Usage**

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

**Arguments**

object            Object of class deltamethod.  
 ...              additional arguments.

**Value**

Returns a vector of estimates.

**Author(s)**

Ivan Jacob Agaloos Pesigan

**Examples**

```
object <- glm(
  formula = vs ~ wt + disp,
  family = "binomial",
  data = mtcars
)
def <- list("exp(wt)", "exp(disp)")
out <- DeltaGeneric(
  object = object,
  def = def,
  alpha = 0.05
)
coef(out)
```

---

coef.diffbetadelta      *Differences of Standardized Regression Slopes*

---

**Description**

Differences of Standardized Regression Slopes

**Usage**

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

**Arguments**

object            Object of class diffbetadelta.  
 ...              additional arguments.

**Value**

Returns a vector of differences of standardized regression slopes.

**Author(s)**

Ivan Jacob Agaloos Pesigan

**Examples**

```
object <- lm(QUALITY ~ NARTIC + PCTGRT + PCTSUPP, data = nas1982)
std <- BetaDelta(object)
diff <- DiffBetaDelta(std)
coef(diff)
```

---

confint.betadelta	<i>Confidence Intervals for Standardized Regression Slopes</i>
-------------------	----------------------------------------------------------------

---

**Description**

Confidence Intervals for Standardized Regression Slopes

**Usage**

```
## S3 method for class 'betadelta'
confint(object, parm = NULL, level = 0.95, ...)
```

**Arguments**

object	Object of class betadelta.
parm	a specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. If missing, all parameters are considered.
level	the confidence level required.
...	additional arguments.

**Value**

Returns a matrix of confidence intervals.

**Author(s)**

Ivan Jacob Agaloos Pesigan

**Examples**

```
object <- lm(QUALITY ~ NARTIC + PCTGRT + PCTSUPP, data = nas1982)
std <- BetaDelta(object)
confint(std, level = 0.95)
```

---

`confint.deltamethod`     *Confidence Intervals*

---

**Description**

Confidence Intervals

**Usage**

```
## S3 method for class 'deltamethod'  
confint(object, parm = NULL, level = 0.95, ...)
```

**Arguments**

<code>object</code>	Object of class <code>deltamethod</code> .
<code>parm</code>	a specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. If missing, all parameters are considered.
<code>level</code>	the confidence level required.
<code>...</code>	additional arguments.

**Value**

Returns a matrix of confidence intervals.

**Author(s)**

Ivan Jacob Agaloos Pesigan

**Examples**

```
object <- glm(  
  formula = vs ~ wt + disp,  
  family = "binomial",  
  data = mtcars  
)  
def <- list("exp(wt)", "exp(disp)")  
out <- DeltaGeneric(  
  object = object,  
  def = def,  
  alpha = 0.05  
)  
confint(out, level = 0.95)
```

---

confint.diffbetadelta *Confidence Intervals for Differences of Standardized Regression Slopes*

---

## Description

Confidence Intervals for Differences of Standardized Regression Slopes

## Usage

```
## S3 method for class 'diffbetadelta'  
confint(object, parm = NULL, level = 0.95, ...)
```

## Arguments

object	Object of class diffbetadelta.
parm	a specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. If missing, all parameters are considered.
level	the confidence level required.
...	additional arguments.

## Value

Returns a matrix of confidence intervals.

## Author(s)

Ivan Jacob Agaloos Pesigan

## Examples

```
object <- lm(QUALITY ~ NARTIC + PCTGRT + PCTSUPP, data = nas1982)  
std <- BetaDelta(object)  
diff <- DiffBetaDelta(std)  
confint(diff)
```



Delta

*Delta Method***Description**

Calculates delta method sampling variance-covariance matrix for a function of parameters using a numerical Jacobian.

**Usage**

```
Delta(
  coef,
  vcov,
  func,
  ...,
  theta = 0,
  alpha = c(0.05, 0.01, 0.001),
  z = TRUE,
  df = NULL
)
```

**Arguments**

<code>coef</code>	Numeric vector. Vector of parameters.
<code>vcov</code>	Numeric matrix. Matrix of sampling variance-covariance matrix of parameters.
<code>func</code>	R function. <ol style="list-style-type: none"> <li>1. The first argument <code>x</code> is the argument <code>coef</code>.</li> <li>2. The function algebraically manipulates <code>coef</code> to return a new numeric vector. It is best to have a named vector as an output.</li> <li>3. The function can take additional named arguments passed using <code>...</code></li> </ol>
<code>...</code>	Additional arguments to pass to <code>func</code> .
<code>theta</code>	Numeric vector. Parameter values when the null hypothesis is true.
<code>alpha</code>	Numeric vector. Significance level/s.
<code>z</code>	Logical. If <code>z = TRUE</code> , use the standard normal distribution. If <code>z = FALSE</code> , use the <code>t</code> distribution.
<code>df</code>	Numeric. Degrees of freedom if <code>z = FALSE</code> .

**Value**

Returns an object of class `deltamethod` which is a list with the following elements:

**call** Function call.

**args** Function arguments.

**coef** Estimates.

**vcov** Sampling variance-covariance matrix.

**jacobian** Jacobian matrix.

**fun** Function used ("Delta").

### Author(s)

Ivan Jacob Agaloos Pesigan

### See Also

Other Delta Method Functions: [DeltaGeneric\(\)](#)

### Examples

```
object <- glm(
  formula = vs ~ wt + disp,
  family = "binomial",
  data = mtcars
)
func <- function(x) {
  y <- exp(x)
  names(y) <- paste0("exp", "(", names(x), ")")
  return(y[-1])
}
Delta(
  coef = coef(object),
  vcov = vcov(object),
  func = func,
  alpha = 0.05
)
```

---

DeltaGeneric

*Delta Method (Generic Object Input)*

---

### Description

Calculates delta method sampling variance-covariance matrix for a function of parameters using a numerical Jacobian.

### Usage

```
DeltaGeneric(
  object,
  def,
  theta = 0,
  alpha = c(0.05, 0.01, 0.001),
  z = TRUE,
  df = NULL
)
```

**Arguments**

<b>object</b>	R object. Fitted model object with <code>coef</code> and <code>vcov</code> methods that return a named vector of estimated parameters and sampling variance-covariance matrix, respectively.
<b>def</b>	List of character strings. A list of defined functions of parameters. The string should be a valid R expression when parsed and should result a single value when evaluated.
<b>theta</b>	Numeric vector. Parameter values when the null hypothesis is true.
<b>alpha</b>	Numeric vector. Significance level/s.
<b>z</b>	Logical. If <code>z = TRUE</code> , use the standard normal distribution. If <code>z = FALSE</code> , use the <code>t</code> distribution.
<b>df</b>	Numeric. Degrees of freedom if <code>z = FALSE</code> .

**Value**

Returns an object of class `deltamethod` which is a list with the following elements:

**call** Function call.

**args** Function arguments.

**coef** Estimates.

**vcov** Sampling variance-covariance matrix.

**jacobian** Jacobian matrix.

**fun** Function used ("DeltaGeneric").

**Author(s)**

Ivan Jacob Agaloos Pesigan

**See Also**

Other Delta Method Functions: [Delta\(\)](#)

**Examples**

```
object <- glm(
  formula = vs ~ wt + disp,
  family = "binomial",
  data = mtcars
)
def <- list("exp(wt)", "exp(disp)")
DeltaGeneric(
  object = object,
  def = def,
  alpha = 0.05
)
```

---

DiffBetaDelta	<i>Estimate Differences of Standardized Slopes and the Corresponding Sampling Covariance Matrix</i>
---------------	-----------------------------------------------------------------------------------------------------

---

**Description**

Estimate Differences of Standardized Slopes and the Corresponding Sampling Covariance Matrix

**Usage**

```
DiffBetaDelta(object, alpha = c(0.05, 0.01, 0.001))
```

**Arguments**

object	Object of class <code>betadelta</code> , that is, the output of the <code>BetaDelta()</code> function.
alpha	Numeric vector. Significance level $\alpha$ .

**Value**

Returns an object of class `diffbetadelta` which is a list with the following elements:

**call** Function call.

**fit** The argument `object`.

**args** Function arguments.

**vcov** Sampling covariance matrix of differences of standardized slopes.

**est** Vector of differences of standardized slopes.

**Author(s)**

Ivan Jacob Agaloos Pesigan

**See Also**

Other Beta Delta Functions: `BetaDelta()`

**Examples**

```
object <- lm(QUALITY ~ NARTIC + PCTGRT + PCTSUPP, data = nas1982)
std <- BetaDelta(object)
diff <- DiffBetaDelta(std)
# Methods -----
print(diff)
summary(diff)
coef(diff)
vcov(diff)
confint(diff, level = 0.95)
```

nas1982

*1982 National Academy of Sciences Doctoral Programs Data***Description**

1982 National Academy of Sciences Doctoral Programs Data

**Usage**

nas1982

**Format**

Ratings of 46 doctoral programs in psychology in the USA with the following variables:

**QUALITY** Program quality ratings.**NFACUL** Number of faculty members in the program.**NGRADS** Number of program graduates.**PCTSUPP** Percentage of program graduates who received support.**PCTGRT** Percent of faculty members holding research grants.**NARTIC** Number of published articles attributed to program faculty member.**PCTPUB** Percent of faculty with one or more published article.**References**

National Research Council. (1982). *An assessment of research-doctorate programs in the United States: Social and behavioral sciences*. doi:10.17226/9781. Reproduced with permission from the National Academy of Sciences, Courtesy of the National Academies Press, Washington, D.C.

print.betadelta

*Print Method for an Object of Class betadelta***Description**

Print Method for an Object of Class betadelta

**Usage**

```
## S3 method for class 'betadelta'
print(x, alpha = NULL, digits = 4, ...)
```

**Arguments**

<code>x</code>	Object of class <code>betadelta</code> .
<code>alpha</code>	Numeric vector. Significance level $\alpha$ . If <code>alpha = NULL</code> , use the argument <code>alpha</code> used in <code>x</code> .
<code>digits</code>	Digits to print.
<code>...</code>	additional arguments.

**Value**

Returns a matrix of standardized regression slopes, standard errors, test statistics, degrees of freedom, p-values, and confidence intervals.

**Author(s)**

Ivan Jacob Agaloos Pesigan

**Examples**

```
object <- lm(QUALITY ~ NARTIC + PCTGRT + PCTSUPP, data = nas1982)
std <- BetaDelta(object)
print(std)
```

---

<code>print.deltamethod</code>	<i>Print Method for an Object of Class <code>deltamethod</code></i>
--------------------------------	---------------------------------------------------------------------

---

**Description**

Print Method for an Object of Class `deltamethod`

**Usage**

```
## S3 method for class 'deltamethod'
print(x, alpha = NULL, digits = 4, ...)
```

**Arguments**

<code>x</code>	Object of class <code>deltamethod</code> .
<code>alpha</code>	Numeric vector. Significance level $\alpha$ . If <code>alpha = NULL</code> , use the argument <code>alpha</code> used in <code>x</code> .
<code>digits</code>	Digits to print.
<code>...</code>	additional arguments.

**Value**

Returns a matrix of coefficients, standard errors, test statistics, degrees of freedom (if `z = FALSE`), p-values, and confidence intervals.

**Author(s)**

Ivan Jacob Agaloos Pesigan

**Examples**

```

object <- glm(
  formula = vs ~ wt + disp,
  family = "binomial",
  data = mtcars
)
def <- list("exp(wt)", "exp(disp)")
out <- DeltaGeneric(
  object = object,
  def = def,
  alpha = 0.05
)
print(out)

```

---

print.diffbetadelta     *Print Method for an Object of Class diffbetadelta*

---

**Description**

Print Method for an Object of Class diffbetadelta

**Usage**

```

## S3 method for class 'diffbetadelta'
print(x, alpha = NULL, digits = 4, ...)

```

**Arguments**

x	Object of class diffbetadelta.
alpha	Numeric vector. Significance level $\alpha$ . If alpha = NULL, use the argument alpha used in x.
digits	Digits to print.
...	additional arguments.

**Value**

Returns a matrix of standardized regression slopes, standard errors, test statistics, degrees of freedom, p-values, and confidence intervals.

**Author(s)**

Ivan Jacob Agaloos Pesigan

**Examples**

```
object <- lm(QUALITY ~ NARTIC + PCTGRT + PCTSUPP, data = nas1982)
std <- BetaDelta(object)
diff <- DiffBetaDelta(std)
print(diff)
```

---

summary.betadelta	<i>Summary Method for an Object of Class betadelta</i>
-------------------	--------------------------------------------------------

---

**Description**

Summary Method for an Object of Class betadelta

**Usage**

```
## S3 method for class 'betadelta'
summary(object, alpha = NULL, digits = 4, ...)
```

**Arguments**

object	Object of class betadelta.
alpha	Numeric vector. Significance level $\alpha$ . If alpha = NULL, use the argument alpha used in object.
digits	Digits to print.
...	additional arguments.

**Value**

Returns a matrix of standardized regression slopes, standard errors, test statistics, degrees of freedom, p-values, and confidence intervals.

**Author(s)**

Ivan Jacob Agaloos Pesigan

**Examples**

```
object <- lm(QUALITY ~ NARTIC + PCTGRT + PCTSUPP, data = nas1982)
std <- BetaDelta(object)
summary(std)
```



---

summary.deltamethod	<i>Summary Method for an Object of Class deltamethod</i>
---------------------	----------------------------------------------------------

---

**Description**

Summary Method for an Object of Class deltamethod

**Usage**

```
## S3 method for class 'deltamethod'  
summary(object, alpha = NULL, digits = 4, ...)
```

**Arguments**

object	Object of class deltamethod.
alpha	Numeric vector. Significance level $\alpha$ . If alpha = NULL, use the argument alpha used in object.
digits	Digits to print.
...	additional arguments.

**Value**

Returns a matrix of standardized regression slopes, standard errors, test statistics, degrees of freedom, p-values, and confidence intervals.

**Author(s)**

Ivan Jacob Agaloos Pesigan

**Examples**

```
object <- glm(  
  formula = vs ~ wt + disp,  
  family = "binomial",  
  data = mtcars  
)  
def <- list("exp(wt)", "exp(disp)")  
out <- DeltaGeneric(  
  object = object,  
  def = def,  
  alpha = 0.05  
)  
summary(out)
```

---

summary.diffbetadelta *Summary Method for an Object of Class diffbetadelta*

---

## Description

Summary Method for an Object of Class diffbetadelta

## Usage

```
## S3 method for class 'diffbetadelta'  
summary(object, alpha = NULL, digits = 4, ...)
```

## Arguments

object	Object of class diffbetadelta.
alpha	Numeric vector. Significance level $\alpha$ . If alpha = NULL, use the argument alpha used in object.
digits	Digits to print.
...	additional arguments.

## Value

Returns a matrix of standardized regression slopes, standard errors, test statistics, degrees of freedom, p-values, and confidence intervals.

## Author(s)

Ivan Jacob Agaloos Pesigan

## Examples

```
object <- lm(QUALITY ~ NARTIC + PCTGRT + PCTSUPP, data = nas1982)  
std <- BetaDelta(object)  
diff <- DiffBetaDelta(std)  
summary(diff)
```

---

vcov.betadelta	<i>Sampling Covariance Matrix of the Standardized Regression Slopes</i>
----------------	-------------------------------------------------------------------------

---

**Description**

Sampling Covariance Matrix of the Standardized Regression Slopes

**Usage**

```
## S3 method for class 'betadelta'
vcov(object, ...)
```

**Arguments**

object	Object of class betadelta.
...	additional arguments.

**Value**

Returns a matrix of the variance-covariance matrix of standardized slopes.

**Author(s)**

Ivan Jacob Agaloos Pesigan

**Examples**

```
object <- lm(QUALITY ~ NARTIC + PCTGRT + PCTSUPP, data = nas1982)
std <- BetaDelta(object)
vcov(std)
```

---

vcov.deltamethod	<i>Sampling Covariance Matrix</i>
------------------	-----------------------------------

---

**Description**

Sampling Covariance Matrix

**Usage**

```
## S3 method for class 'deltamethod'
vcov(object, ...)
```

**Arguments**

object            Object of class `deltamethod`.  
 ...              additional arguments.

**Value**

Returns a matrix of the variance-covariance matrix.

**Author(s)**

Ivan Jacob Agaloos Pesigan

**Examples**

```
object <- glm(
  formula = vs ~ wt + disp,
  family = "binomial",
  data = mtcars
)
def <- list("exp(wt)", "exp(disp)")
out <- DeltaGeneric(
  object = object,
  def = def,
  alpha = 0.05
)
vcov(out)
```

---

vcov.diffbetadelta	<i>Sampling Covariance Matrix of Differences of Standardized Regression Slopes</i>
--------------------	------------------------------------------------------------------------------------

---

**Description**

Sampling Covariance Matrix of Differences of Standardized Regression Slopes

**Usage**

```
## S3 method for class 'diffbetadelta'
vcov(object, ...)
```

**Arguments**

object            Object of class `diffbetadelta`.  
 ...              additional arguments.

**Value**

Returns a matrix of the variance-covariance matrix of differences of standardized regression slopes.

**Author(s)**

Ivan Jacob Agaloos Pesigan

**Examples**

```
object <- lm(QUALITY ~ NARTIC + PCTGRT + PCTSUPP, data = nas1982)
std <- BetaDelta(object)
diff <- DiffBetaDelta(std)
vcov(diff)
```

# Index

- \* **Beta Delta Functions**

- BetaDelta, [2](#)
  - DiffBetaDelta, [12](#)

- \* **Delta Method Functions**

- Delta, [9](#)
  - DeltaGeneric, [10](#)

- \* **betaDelta**

- BetaDelta, [2](#)
  - DiffBetaDelta, [12](#)

- \* **data**

- nas1982, [13](#)

- \* **deltaMethod**

- Delta, [9](#)
  - DeltaGeneric, [10](#)

- \* **diff**

- DiffBetaDelta, [12](#)

- \* **methods**

- coef.betadelta, [4](#)
  - coef.deltamethod, [4](#)
  - coef.diffbetadelta, [5](#)
  - confint.betadelta, [6](#)
  - confint.deltamethod, [7](#)
  - confint.diffbetadelta, [8](#)
  - print.betadelta, [13](#)
  - print.deltamethod, [14](#)
  - print.diffbetadelta, [15](#)
  - summary.betadelta, [16](#)
  - summary.deltamethod, [17](#)
  - summary.diffbetadelta, [18](#)
  - vcov.betadelta, [19](#)
  - vcov.deltamethod, [19](#)
  - vcov.diffbetadelta, [20](#)

- \* **std**

- BetaDelta, [2](#)

BetaDelta, [2](#), [12](#)

BetaDelta(), [12](#)

coef.betadelta, [4](#)

coef.deltamethod, [4](#)

coef.diffbetadelta, [5](#)

confint.betadelta, [6](#)

confint.deltamethod, [7](#)

confint.diffbetadelta, [8](#)

Delta, [9](#), [11](#)

DeltaGeneric, [10](#), [10](#)

DiffBetaDelta, [3](#), [12](#)

nas1982, [13](#)

print.betadelta, [13](#)

print.deltamethod, [14](#)

print.diffbetadelta, [15](#)

summary.betadelta, [16](#)

summary.deltamethod, [17](#)

summary.diffbetadelta, [18](#)

vcov.betadelta, [19](#)

vcov.deltamethod, [19](#)

vcov.diffbetadelta, [20](#)