betaDelta: Internal Tests

Ivan Jacob Agaloos Pesigan

Tests

```
#> test-betaDelta-beta-delta-adf
#> Test passed
#> Test passed

#> test-betaDelta-beta-delta-methods

#> Call:
#> BetaDelta(object = object, type = "mvn")
#>
#> Standardized regression slopes with MVN standard errors:
#>            est     se      t df     p   0.05%    0.5%   2.5%  97.5%  99.5%
#> NARTIC  0.4951 0.0759 6.5272 42 0.000  0.2268 0.2905 0.3421 0.6482 0.6998
#> PCTGRT  0.3915 0.0770 5.0824 42 0.000  0.1190 0.1837 0.2360 0.5469 0.5993
#> PCTSUPP 0.2632 0.0747 3.5224 42 0.001 -0.0011 0.0616 0.1124 0.4141 0.4649
#>         99.95%
#> NARTIC  0.7635
#> PCTGRT  0.6640
#> PCTSUPP 0.5276
#> Call:
#> BetaDelta(object = object, type = "adf")
#>
#> Standardized regression slopes with ADF standard errors:
#>            est     se      t df      p   0.05%    0.5%   2.5%  97.5%  99.5%
#> NARTIC  0.4951 0.0674 7.3490 42 0.0000  0.2568 0.3134 0.3592 0.6311 0.6769
#> PCTGRT  0.3915 0.0710 5.5164 42 0.0000  0.1404 0.2000 0.2483 0.5347 0.5830
#> PCTSUPP 0.2632 0.0769 3.4231 42 0.0014 -0.0088 0.0558 0.1081 0.4184 0.4707
#>         99.95%
#> NARTIC  0.7335
#> PCTGRT  0.6426
#> PCTSUPP 0.5353
#> Call:
#> BetaDelta(object = object, type = "mvn")
#>
#> Standardized regression slopes with MVN standard errors:
#>            est     se      t df p  0.05%   0.5%   2.5%  97.5%  99.5% 99.95%
```

```
#> NARTIC 0.7622 0.0618 12.3341 44 0 0.5443 0.5958 0.6376 0.8867 0.9285   0.98
#> Call:
#> BetaDelta(object = object, type = "adf")
#>
#> Standardized regression slopes with ADF standard errors:
#>           est     se      t df p  0.05%   0.5%   2.5% 97.5% 99.5% 99.95%
#> NARTIC 0.7622 0.0604 12.625 44 0 0.5493 0.5996 0.6405 0.8838 0.9247  0.975
#> Test passed

#> test-betaDelta-beta-delta-mvn

#> Test passed
#> Test passed

#> test-betaDelta-diff-beta-delta-methods

#> Call:
#> DiffBetaDelta(object = BetaDelta(object, type = "mvn"))
#>
#> Difference between standardized regression coefficients with MVN standard errors:
#>                   est     se      z      p   0.05%    0.5%    2.5%  97.5%
#> NARTIC-PCTGRT  0.1037 0.1357 0.7640 0.4449 -0.3428 -0.2458 -0.1623 0.3696
#> NARTIC-PCTSUPP 0.2319 0.1252 1.8524 0.0640 -0.1800 -0.0906 -0.0135 0.4773
#> PCTGRT-PCTSUPP 0.1282 0.1227 1.0451 0.2960 -0.2755 -0.1878 -0.1123 0.3688
#>                 99.5% 99.95%
#> NARTIC-PCTGRT  0.4531 0.5501
#> NARTIC-PCTSUPP 0.5544 0.6438
#> PCTGRT-PCTSUPP 0.4443 0.5320
#> Call:
#> DiffBetaDelta(object = BetaDelta(object, type = "adf"))
#>
#> Difference between standardized regression coefficients with ADF standard errors:
#>                   est     se      z      p   0.05%    0.5%    2.5%  97.5%
#> NARTIC-PCTGRT  0.1037 0.1212 0.8555 0.3923 -0.2950 -0.2084 -0.1338 0.3411
#> NARTIC-PCTSUPP 0.2319 0.1181 1.9642 0.0495 -0.1566 -0.0722  0.0005 0.4633
#> PCTGRT-PCTSUPP 0.1282 0.1215 1.0555 0.2912 -0.2716 -0.1847 -0.1099 0.3664
#>                 99.5% 99.95%
#> NARTIC-PCTGRT  0.4158 0.5024
#> NARTIC-PCTSUPP 0.5360 0.6204
#> PCTGRT-PCTSUPP 0.4412 0.5281
#> Test passed

#> test-betaDelta-diff-beta-delta

#> Test passed
#> Test passed
#> Test passed
#> Test passed
```

```
#> test-betaDelta

#> Test passed

#> test-deltaMethod-delta-generic

#> Test passed
#> Test passed
#> Test passed
#> Call:
#> DeltaGeneric(object = object, def = def, alpha = 0.05)
#>              est     se       z      p    2.5%   97.5%
#> exp(wt)   5.0853 7.5805  0.6708 0.5023 -9.7723 19.9429
#> exp(disp) 0.9662 0.0148 65.0838 0.0000  0.9371  0.9952
#> Call:
#> DeltaGeneric(object = object, def = def, alpha = 0.05)
#> Call:
#> DeltaGeneric(object = object, def = list("exp(wt)"), alpha = 0.05,
#>     z = FALSE, df = 30)
#>           est     se      t df      p     2.5%   97.5%
#> exp(wt) 5.0853 7.5805 0.6708 30 0.5075 -10.3962 20.5668
#> Call:
#> DeltaGeneric(object = object, def = list("exp(wt)"), alpha = 0.05,
#>     z = FALSE, df = 30)
#> Test passed

#> test-deltaMethod-delta

#> Test passed
#> Test passed
#> Test passed
#> Call:
#> Delta(coef = coef, vcov = vcov, func = func, alpha = 0.05)
#>              est     se       z      p    2.5%   97.5%
#> exp(wt)   5.0853 7.5805  0.6708 0.5023 -9.7723 19.9429
#> exp(disp) 0.9662 0.0148 65.0838 0.0000  0.9371  0.9952
#> Call:
#> Delta(coef = coef, vcov = vcov, func = func, alpha = 0.05)
#> Call:
#> Delta(coef = coef[2], vcov = vcov[2, 2, drop = FALSE], func = func,
#>     alpha = 0.05, z = FALSE, df = 30)
#>           est     se      t df      p     2.5%   97.5%
#> exp(wt) 5.0853 7.5805 0.6708 30 0.5075 -10.3962 20.5668
#> Call:
#> Delta(coef = coef[2], vcov = vcov[2, 2, drop = FALSE], func = func,
#>     alpha = 0.05, z = FALSE, df = 30)
#> Test passed
```

```
#> test-zzz-coverage

#> Test passed
#> Test passed
#> Test passed
#> [[1]]
#> [[1]][[1]]
#> [[1]][[1]]$value
#> [[1]][[1]]$value[[1]]
#> [1] TRUE
#>
#>
#> [[1]][[1]]$visible
#> [1] TRUE
#>
#>
#> [[1]][[2]]
#> [[1]][[2]]$value
#> [[1]][[2]]$value[[1]]
#> [1] TRUE
#>
#>
#> [[1]][[2]]$visible
#> [1] TRUE
#>
#>
#> [[1]][[3]]
#> [[1]][[3]]$value
#> [[1]][[3]]$value[[1]]
#> [1] TRUE
#>
#>
#> [[1]][[3]]$visible
#> [1] TRUE
#>
#>
#> [[1]][[4]]
#> [[1]][[4]]$value
#> [[1]][[4]]$value[[1]]
#> [1] TRUE
#>
#>
#> [[1]][[4]]$visible
#> [1] TRUE
#>
#>
```

```
#> [[1]][[5]]
#> [[1]][[5]]$value
#> [[1]][[5]]$value[[1]]
#> [1] TRUE
#>
#>
#> [[1]][[5]]$visible
#> [1] TRUE
#>
#>
#> [[1]][[6]]
#> [[1]][[6]]$value
#> [[1]][[6]]$value[[1]]
#> [1] TRUE
#>
#>
#> [[1]][[6]]$visible
#> [1] TRUE
#>
#>
#> [[1]][[7]]
#> [[1]][[7]]$value
#> [[1]][[7]]$value[[1]]
#> [1] TRUE
#>
#>
#> [[1]][[7]]$visible
#> [1] TRUE
#>
#>
#> [[1]][[8]]
#> [[1]][[8]]$value
#> [[1]][[8]]$value[[1]]
#> [1] TRUE
#>
#>
#> [[1]][[8]]$visible
#> [1] TRUE
#>
#>
#> [[1]][[9]]
#> [[1]][[9]]$value
#> [[1]][[9]]$value[[1]]
#> [1] TRUE
#>
```

```
#>
#> [[1]][[9]]$visible
#> [1] TRUE
```

Environment

```
ls()

#> [1] "nas1982" "root"
```

Class

```
#> [[1]]
#> [1] "data.frame"
#>
#> [[2]]
#> [1] "root_criterion"
```

References

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*, *58*(6), 1183–1186. https://doi.org/10.1080/00273171.2023.2201277

R Core Team. (2024). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. Vienna, Austria. https://www.R-project.org/