betaSandwich: Internal Tests

Ivan Jacob Agaloos Pesigan

Tests

```
#> test-betaSandwich-beta-sandwich-adf
#> Test passed
#> Call:
#> BetaADF(object = object)
#>
#> Standardized regression slopes with MVN standard errors:
#> Call:
#> BetaADF(object = object)
#>
#> Standardized regression slopes with MVN standard errors:
#> Test passed
```

```
#> test-betaSandwich-beta-sandwich-hc
```

```
#> Test passed
#> Call:
#> BetaHC(object = object, type = "hc0")
#>
#> Standardized regression slopes with HC0 standard errors:
#> Call:
#> BetaHC(object = object, type = "hc0")
#>
#> Standardized regression slopes with HC0 standard errors:
#> Test passed
#> Call:
#> BetaHC(object = object, type = "hc1")
#>
#> Standardized regression slopes with HC1 standard errors:
#> Call:
#> BetaHC(object = object, type = "hc1")
#>
#> Standardized regression slopes with HC1 standard errors:
#> Test passed
#> Call:
#> BetaHC(object = object, type = "hc2")
```

```
#>
#> Standardized regression slopes with HC2 standard errors:
#> Call:
#> BetaHC(object = object, type = "hc2")
#>
#> Standardized regression slopes with HC2 standard errors:
#> Test passed
#> Call:
#> BetaHC(object = object, type = "hc3")
#>
#> Standardized regression slopes with HC3 standard errors:
#> Call:
#> BetaHC(object = object, type = "hc3")
#>
#> Standardized regression slopes with HC3 standard errors:
#> Test passed
#> Call:
#> BetaHC(object = object, type = "hc4")
#>
#> Standardized regression slopes with HC4 standard errors:
#> Call:
#> BetaHC(object = object, type = "hc4")
#>
#> Standardized regression slopes with HC4 standard errors:
#> Test passed
#> Call:
#> BetaHC(object = object, type = "hc4m")
#>
#> Standardized regression slopes with HC4M standard errors:
#> Call:
#> BetaHC(object = object, type = "hc4m")
#>
#> Standardized regression slopes with HC4M standard errors:
#> Test passed
#> Call:
#> BetaHC(object = object, type = "hc5")
#>
#> Standardized regression slopes with HC5 standard errors:
#> Call:
#> BetaHC(object = object, type = "hc5")
#>
#> Standardized regression slopes with HC5 standard errors:
#> Test passed
```

```
#> test-betaSandwich-beta-sandwich-methods
```

```
#> Call:
```

```
#> BetaHC(object = object)
#> 
#> Standardized regression slopes with HC3 standard errors:
#>            est     se      t df      p   2.5%  97.5%
#> NARTIC  0.4951 0.0786 6.3025 42 0.0000 0.3366 0.6537
#> PCTGRT  0.3915 0.0818 4.7831 42 0.0000 0.2263 0.5567
#> PCTSUPP 0.2632 0.0855 3.0786 42 0.0037 0.0907 0.4358
#> Call:
#> BetaHC(object = object)
#> 
#> Standardized regression slopes with HC3 standard errors:
#> Call:
#> BetaN(object = object)
#> 
#> Standardized regression slopes with MVN standard errors:
#>            est     se      t df     p   2.5%  97.5%
#> NARTIC  0.4951 0.0759 6.5272 42 0.000 0.3421 0.6482
#> PCTGRT  0.3915 0.0770 5.0824 42 0.000 0.2360 0.5469
#> PCTSUPP 0.2632 0.0747 3.5224 42 0.001 0.1124 0.4141
#> Call:
#> BetaN(object = object)
#> 
#> Standardized regression slopes with MVN standard errors:
#> Call:
#> BetaADF(object = object)
#> 
#> Standardized regression slopes with MVN standard errors:
#>            est     se      t df      p   2.5%  97.5%
#> NARTIC  0.4951 0.0674 7.3490 42 0.0000 0.3592 0.6311
#> PCTGRT  0.3915 0.0710 5.5164 42 0.0000 0.2483 0.5347
#> PCTSUPP 0.2632 0.0769 3.4231 42 0.0014 0.1081 0.4184
#> Call:
#> BetaADF(object = object)
#> 
#> Standardized regression slopes with MVN standard errors:
#> Call:
#> BetaHC(object = object)
#> 
#> Standardized regression slopes with HC3 standard errors:
#>           est     se       t df p   2.5%  97.5%
#> NARTIC 0.7622 0.0645 11.8222 44 0 0.6322 0.8921
#> Call:
#> BetaHC(object = object)
#> 
#> Standardized regression slopes with HC3 standard errors:
```

```
#> Call:
#> BetaN(object = object)
#>
#> Standardized regression slopes with MVN standard errors:
#>           est     se       t df p   2.5%  97.5%
#> NARTIC 0.7622 0.0618 12.3341 44 0 0.6376 0.8867
#> Call:
#> BetaN(object = object)
#>
#> Standardized regression slopes with MVN standard errors:
#> Call:
#> BetaADF(object = object)
#>
#> Standardized regression slopes with MVN standard errors:
#>           est     se      t df p   2.5%  97.5%
#> NARTIC 0.7622 0.0604 12.625 44 0 0.6405 0.8838
#> Call:
#> BetaADF(object = object)
#>
#> Standardized regression slopes with MVN standard errors:

#> test-betaSandwich-beta-sandwich-mvn

#> Test passed
#> Call:
#> BetaN(object = object)
#>
#> Standardized regression slopes with MVN standard errors:
#> Call:
#> BetaN(object = object)
#>
#> Standardized regression slopes with MVN standard errors:
#> Test passed

#> test-betaSandwich-diff-beta-sandwich-methods

#> Call:
#> DiffBetaSandwich(object = BetaN(object))
#>
#> Difference between standardized regression coefficients with MVN standard errors:
#>                   est     se      z      p    2.5%  97.5%
#> NARTIC-PCTGRT  0.1037 0.1357 0.7640 0.4449 -0.1623 0.3696
#> NARTIC-PCTSUPP 0.2319 0.1252 1.8524 0.0640 -0.0135 0.4773
#> PCTGRT-PCTSUPP 0.1282 0.1227 1.0451 0.2960 -0.1123 0.3688
#> Call:
#> DiffBetaSandwich(object = BetaN(object))
#>
```

```
#> Difference between standardized regression coefficients with MVN standard errors:
#> Call:
#> DiffBetaSandwich(object = BetaADF(object))
#>
#> Difference between standardized regression coefficients with MVN standard errors:
#>                   est     se      z      p    2.5%  97.5%
#> NARTIC-PCTGRT  0.1037 0.1212 0.8555 0.3923 -0.1338 0.3411
#> NARTIC-PCTSUPP 0.2319 0.1181 1.9642 0.0495  0.0005 0.4633
#> PCTGRT-PCTSUPP 0.1282 0.1215 1.0555 0.2912 -0.1099 0.3664
#> Call:
#> DiffBetaSandwich(object = BetaADF(object))
#>
#> Difference between standardized regression coefficients with MVN standard errors:
#> Call:
#> DiffBetaSandwich(object = BetaHC(object, type = "hc0"))
#>
#> Difference between standardized regression coefficients with HC0 standard errors:
#>                   est     se      z      p    2.5%  97.5%
#> NARTIC-PCTGRT  0.1037 0.1201 0.8629 0.3882 -0.1318 0.3391
#> NARTIC-PCTSUPP 0.2319 0.1169 1.9840 0.0473  0.0028 0.4610
#> PCTGRT-PCTSUPP 0.1282 0.1201 1.0674 0.2858 -0.1072 0.3637
#> Call:
#> DiffBetaSandwich(object = BetaHC(object, type = "hc0"))
#>
#> Difference between standardized regression coefficients with HC0 standard errors:
#> Call:
#> DiffBetaSandwich(object = BetaHC(object, type = "hc1"))
#>
#> Difference between standardized regression coefficients with HC1 standard errors:
#>                   est     se      z      p    2.5%  97.5%
#> NARTIC-PCTGRT  0.1037 0.1257 0.8245 0.4097 -0.1427 0.3501
#> NARTIC-PCTSUPP 0.2319 0.1223 1.8958 0.0580 -0.0078 0.4716
#> PCTGRT-PCTSUPP 0.1282 0.1257 1.0199 0.3078 -0.1182 0.3747
#> Call:
#> DiffBetaSandwich(object = BetaHC(object, type = "hc1"))
#>
#> Difference between standardized regression coefficients with HC1 standard errors:
#> Call:
#> DiffBetaSandwich(object = BetaHC(object, type = "hc2"))
#>
#> Difference between standardized regression coefficients with HC2 standard errors:
#>                   est     se      z      p    2.5%  97.5%
#> NARTIC-PCTGRT  0.1037 0.1302 0.7960 0.4260 -0.1516 0.3589
#> NARTIC-PCTSUPP 0.2319 0.1240 1.8704 0.0614 -0.0111 0.4749
#> PCTGRT-PCTSUPP 0.1282 0.1284 0.9990 0.3178 -0.1234 0.3798
#> Call:
```

```
#> DiffBetaSandwich(object = BetaHC(object, type = "hc2"))
#>
#> Difference between standardized regression coefficients with HC2 standard errors:
#> Call:
#> DiffBetaSandwich(object = BetaHC(object, type = "hc3"))
#>
#> Difference between standardized regression coefficients with HC3 standard errors:
#>                    est     se      z      p    2.5%  97.5%
#> NARTIC-PCTGRT  0.1037 0.1417 0.7316 0.4644 -0.1741 0.3814
#> NARTIC-PCTSUPP 0.2319 0.1318 1.7595 0.0785 -0.0264 0.4902
#> PCTGRT-PCTSUPP 0.1282 0.1375 0.9329 0.3509 -0.1412 0.3977
#> Call:
#> DiffBetaSandwich(object = BetaHC(object, type = "hc3"))
#>
#> Difference between standardized regression coefficients with HC3 standard errors:
#> Call:
#> DiffBetaSandwich(object = BetaHC(object, type = "hc4"))
#>
#> Difference between standardized regression coefficients with HC4 standard errors:
#>                    est     se      z      p    2.5%  97.5%
#> NARTIC-PCTGRT  0.1037 0.1452 0.7138 0.4753 -0.1809 0.3883
#> NARTIC-PCTSUPP 0.2319 0.1296 1.7892 0.0736 -0.0221 0.4859
#> PCTGRT-PCTSUPP 0.1282 0.1361 0.9420 0.3462 -0.1386 0.3951
#> Call:
#> DiffBetaSandwich(object = BetaHC(object, type = "hc4"))
#>
#> Difference between standardized regression coefficients with HC4 standard errors:
#> Call:
#> DiffBetaSandwich(object = BetaHC(object, type = "hc4m"))
#>
#> Difference between standardized regression coefficients with HC4M standard errors:
#>                    est     se      z      p    2.5%  97.5%
#> NARTIC-PCTGRT  0.1037 0.1465 0.7077 0.4791 -0.1834 0.3907
#> NARTIC-PCTSUPP 0.2319 0.1338 1.7331 0.0831 -0.0304 0.4941
#> PCTGRT-PCTSUPP 0.1282 0.1406 0.9123 0.3616 -0.1473 0.4037
#> Call:
#> DiffBetaSandwich(object = BetaHC(object, type = "hc4m"))
#>
#> Difference between standardized regression coefficients with HC4M standard errors:
#> Call:
#> DiffBetaSandwich(object = BetaHC(object, type = "hc5"))
#>
#> Difference between standardized regression coefficients with HC5 standard errors:
#>                    est     se      z      p    2.5%  97.5%
#> NARTIC-PCTGRT  0.1037 0.1312 0.7899 0.4296 -0.1536 0.3609
#> NARTIC-PCTSUPP 0.2319 0.1227 1.8906 0.0587 -0.0085 0.4723
#> PCTGRT-PCTSUPP 0.1282 0.1274 1.0067 0.3141 -0.1214 0.3779
#> Call:
#> DiffBetaSandwich(object = BetaHC(object, type = "hc5"))
#>
#> Difference between standardized regression coefficients with HC5 standard errors:
```

#> test-betaSandwich-diff-beta-sandwich

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

#> test-betaSandwich-r-sq-beta-sandwich-methods

```
#> Call:
#> RSqBetaSandwich(object = BetaN(object))
#>
#> Multiple correlation with MVN standard errors:
#>        est     se      t df      p    2.5%  97.5%
#> rsq 0.8045 4.1345 0.1946 42 0.8467 -7.5393 9.1483
#> adj 0.7906 4.4299 0.1785 42 0.8592 -8.1492 9.7304
#> Call:
#> RSqBetaSandwich(object = BetaN(object))
#>
#> Multiple correlation with MVN standard errors:
#> Call:
#> RSqBetaSandwich(object = BetaADF(object))
#>
#> Multiple correlation with MVN standard errors:
#>        est     se      t df      p    2.5%  97.5%
#> rsq 0.8045 3.6172 0.2224 42 0.8251 -6.4953 8.1044
#> adj 0.7906 3.8756 0.2040 42 0.8394 -7.0307 8.6118
#> Call:
#> RSqBetaSandwich(object = BetaADF(object))
#>
#> Multiple correlation with MVN standard errors:
#> Call:
#> RSqBetaSandwich(object = BetaHC(object))
#>
#> Multiple correlation with HC3 standard errors:
#>        est     se      t df      p    2.5%  97.5%
#> rsq 0.8045 3.9483 0.2038 42 0.8395 -7.1635 8.7725
#> adj 0.7906 4.2303 0.1869 42 0.8527 -7.7466 9.3277
#> Call:
#> RSqBetaSandwich(object = BetaHC(object))
#>
#> Multiple correlation with HC3 standard errors:
#> Call:
#> RSqBetaSandwich(object = BetaN(object))
#>
#> Multiple correlation with MVN standard errors:
#>        est     se      t df     p     2.5%   97.5%
#> rsq 0.5809 8.8646 0.0655 44 0.948 -17.2845 18.4463
#> adj 0.5714 9.0661 0.0630 44 0.950 -17.7001 18.8428
#> Call:
#> RSqBetaSandwich(object = BetaN(object))
#>
#> Multiple correlation with MVN standard errors:
#> Call:
#> RSqBetaSandwich(object = BetaADF(object))
#>
#> Multiple correlation with MVN standard errors:
#>        est     se      t df      p     2.5%   97.5%
#> rsq 0.5809 8.5690 0.0678 44 0.9463 -16.6887 17.8505
#> adj 0.5714 8.7637 0.0652 44 0.9483 -17.0907 18.2335
#> Call:
#> RSqBetaSandwich(object = BetaADF(object))
#>
#> Multiple correlation with MVN standard errors:
#> Call:
#> RSqBetaSandwich(object = BetaHC(object))
#>
#> Multiple correlation with HC3 standard errors:
#>        est     se      t df      p     2.5%   97.5%
#> rsq 0.5809 8.9208 0.0651 44 0.9484 -17.3978 18.5596
#> adj 0.5714 9.1235 0.0626 44 0.9503 -17.8159 18.9586
#> Call:
```

```
#> RSqBetaSandwich(object = BetaHC(object))
#>
#> Multiple correlation with HC3 standard errors:

#> test-betaSandwich-r-sq-beta-sandwich

#> Call:
#> RSqBetaSandwich(object = BetaN(object))
#>
#> Multiple correlation with MVN standard errors:
#> Test passed
#> Call:
#> RSqBetaSandwich(object = BetaADF(object))
#>
#> Multiple correlation with MVN standard errors:
#> Test passed
#> Call:
#> RSqBetaSandwich(object = BetaHC(object))
#>
#> Multiple correlation with HC3 standard errors:
#> Test passed
#> Call:
#> RSqBetaSandwich(object = BetaN(object))
#>
#> Multiple correlation with MVN standard errors:
#> Test passed
#> Call:
#> RSqBetaSandwich(object = BetaADF(object))
#>
#> Multiple correlation with MVN standard errors:
#> Test passed
#> Call:
#> RSqBetaSandwich(object = BetaHC(object))
#>
#> Multiple correlation with HC3 standard errors:
#> Test passed

#> test-zzz-coverage

#>               beta1    beta2    beta3 sigmasq   sigmax1x1  sigmax2x1  sigmax3x1
#> sigmaysq   909.1981 257.2976 276.0367       1 0.007091036 0.03637752 0.01896371
#> sigmayx1  3507.1691 471.2058 510.5430       0 0.084208291 0.21599726 0.11260003
#> sigmayx2   471.2058 333.2295 150.9121       0 0.000000000 0.08420829 0.00000000
#> sigmayx3   510.5430 150.9121 554.4386       0 0.000000000 0.00000000 0.08420829
#> sigmax1x1    0.0000   0.0000   0.0000       0 1.000000000 0.00000000 0.00000000
#> sigmax2x1    0.0000   0.0000   0.0000       0 0.000000000 1.00000000 0.00000000
#> sigmax3x1    0.0000   0.0000   0.0000       0 0.000000000 0.00000000 1.00000000
#> sigmax2x2    0.0000   0.0000   0.0000       0 0.000000000 0.00000000 0.00000000
#> sigmax3x2    0.0000   0.0000   0.0000       0 0.000000000 0.00000000 0.00000000
#> sigmax3x3    0.0000   0.0000   0.0000       0 0.000000000 0.00000000 0.00000000
#>            sigmax2x2 sigmax3x2  sigmax3x3
#> sigmaysq  0.04665482 0.0486426 0.01267877
#> sigmayx1  0.00000000 0.0000000 0.00000000
#> sigmayx2  0.21599726 0.1126000 0.00000000
#> sigmayx3  0.00000000 0.2159973 0.11260003
#> sigmax1x1 0.00000000 0.0000000 0.00000000
#> sigmax2x1 0.00000000 0.0000000 0.00000000
#> sigmax3x1 0.00000000 0.0000000 0.00000000
#> sigmax2x2 1.00000000 0.0000000 0.00000000
#> sigmax3x2 0.00000000 1.0000000 0.00000000
#> sigmax3x3 0.00000000 0.0000000 1.00000000
#>               beta1    beta2    beta3 sigmasq
#> sigmaysq   909.1981 257.2976 276.0367       1
#> sigmayx1  3507.1691 471.2058 510.5430       0
#> sigmayx2   471.2058 333.2295 150.9121       0
#> sigmayx3   510.5430 150.9121 554.4386       0
#> sigmax1x1    0.0000   0.0000   0.0000       0
#> sigmax2x1    0.0000   0.0000   0.0000       0
#> sigmax3x1    0.0000   0.0000   0.0000       0
#> sigmax2x2    0.0000   0.0000   0.0000       0
#> sigmax3x2    0.0000   0.0000   0.0000       0
#> sigmax3x3    0.0000   0.0000   0.0000       0
#>               beta1    beta2    beta3       rsq   sigmax1x1  sigmax2x1
#> sigmaysq   909.1981 257.2976 276.0367 -126.0843 0.007091036 0.03637752
#> sigmayx1  3507.1691 471.2058 510.5430    0.0000 0.084208291 0.21599726
#> sigmayx2   471.2058 333.2295 150.9121    0.0000 0.000000000 0.08420829
#> sigmayx3   510.5430 150.9121 554.4386    0.0000 0.000000000 0.00000000
#> sigmax1x1    0.0000   0.0000   0.0000    0.0000 1.000000000 0.00000000
#> sigmax2x1    0.0000   0.0000   0.0000    0.0000 0.000000000 1.00000000
#> sigmax3x1    0.0000   0.0000   0.0000    0.0000 0.000000000 0.00000000
#> sigmax2x2    0.0000   0.0000   0.0000    0.0000 0.000000000 0.00000000
#> sigmax3x2    0.0000   0.0000   0.0000    0.0000 0.000000000 0.00000000
#> sigmax3x3    0.0000   0.0000   0.0000    0.0000 0.000000000 0.00000000
#>            sigmax3x1  sigmax2x2 sigmax3x2  sigmax3x3
#> sigmaysq  0.01896371 0.04665482 0.0486426 0.01267877
#> sigmayx1  0.11260003 0.00000000 0.0000000 0.00000000
#> sigmayx2  0.00000000 0.21599726 0.1126000 0.00000000
#> sigmayx3  0.08420829 0.00000000 0.2159973 0.11260003
#> sigmax1x1 0.00000000 0.00000000 0.0000000 0.00000000
#> sigmax2x1 0.00000000 0.00000000 0.0000000 0.00000000
#> sigmax3x1 1.00000000 0.00000000 0.0000000 0.00000000
#> sigmax2x2 0.00000000 1.00000000 0.0000000 0.00000000
#> sigmax3x2 0.00000000 0.00000000 1.0000000 0.00000000
#> sigmax3x3 0.00000000 0.00000000 0.0000000 1.00000000
#>               beta1    beta2    beta3       rsq
#> sigmaysq   909.1981 257.2976 276.0367 -126.0843
#> sigmayx1  3507.1691 471.2058 510.5430    0.0000
#> sigmayx2   471.2058 333.2295 150.9121    0.0000
#> sigmayx3   510.5430 150.9121 554.4386    0.0000
#> sigmax1x1    0.0000   0.0000   0.0000    0.0000
#> sigmax2x1    0.0000   0.0000   0.0000    0.0000
#> sigmax3x1    0.0000   0.0000   0.0000    0.0000
#> sigmax2x2    0.0000   0.0000   0.0000    0.0000
#> sigmax3x2    0.0000   0.0000   0.0000    0.0000
#> sigmax3x3    0.0000   0.0000   0.0000    0.0000
#> [[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]]
#>      2.5%     97.5%
#> 0.6404985 0.8838331
#>
#>
#> [[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]]
#>                        2.5%     97.5%
#> NARTIC-PCTGRT  -0.153558897 0.3608717
#> NARTIC-PCTSUPP -0.008509607 0.4723045
#> PCTGRT-PCTSUPP -0.121440586 0.3779227
#> 
#> 
#> [[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]]
#>          2.5%    97.5%
#> rsq -17.39776 18.55955
#> adj -17.81589 18.95863
#> 
#> 
#> [[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]]
#>               beta1    beta2    beta3       rsq
#> sigmaysq   909.1981 257.2976 276.0367 -126.0843
#> sigmayx1  3507.1691 471.2058 510.5430    0.0000
#> sigmayx2   471.2058 333.2295 150.9121    0.0000
#> sigmayx3   510.5430 150.9121 554.4386    0.0000
#> sigmax1x1    0.0000   0.0000   0.0000    0.0000
#> sigmax2x1    0.0000   0.0000   0.0000    0.0000
#> sigmax3x1    0.0000   0.0000   0.0000    0.0000
#> sigmax2x2    0.0000   0.0000   0.0000    0.0000
#> sigmax3x2    0.0000   0.0000   0.0000    0.0000
#> sigmax3x3    0.0000   0.0000   0.0000    0.0000
#>
#>
#> [[1]][[9]]$visible
#> [1] TRUE
```

Environment

```
ls()

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

Class

```
#> [[1]]
#> [1] "data.frame"
#>
#> [[2]]
#> [1] "root_criterion"
#>
#> [[3]]
#> [1] "character"
```

References

R Core Team. (2023). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. Vienna, Austria. https://www.R-project.org/