

R version 4.4.0 (2024-04-24) -- "Puppy Cup"  
Copyright (C) 2024 The R Foundation for Statistical Computing  
Platform: aarch64-apple-darwin20

R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions.  
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.

[R.app GUI 1.80 (8376) aarch64-apple-darwin20]

[History restored from /Users/alperkaragol/.Rapp.history]

```
> library(generalCorr)
Loading required package: np
Nonparametric Kernel Methods for Mixed Datatypes (version 0.60-17)
[vignette("np_faq",package="np") provides answers to frequently asked questions]
[vignette("np",package="np") an overview]
[vignette("entropy_np",package="np") an overview of entropy-based methods]
Loading required package: xtable
Loading required package: meboot
Loading required package: dynlm
Loading required package: zoo

Attaching package: 'zoo'

The following objects are masked from 'package:base':

    as.Date, as.Date.numeric

Loading required package: nlme
Loading required package: tdigest
Loading required package: hdrdce
This is hdrdce 3.4
Loading required package: psych
Loading required package: lattice
> R <- c(3.927,8.273,3.050,3.069,2.446,17.692,13.176,0.448,4.202,0.745,0.468,0.590,0.413,3.228,1.446,0.665)
> M <- c(-131.9,-1075,-760.6,-156.2,-226.7,-150,-144.7,-482,-515.6,-456.4,-1290.2,-168,-177.5, 226.3,- 324.3,- 991)
> L <- c(369,3717,2811,566,784,488,446,1313,1351,1353,4608,629,607, 726, 695, 2471)
> E <- c(97.27, 1013.12, 643.41, 160.33, 206.26, 72.09, 99.32, 294.18, 328.73, 264.82, 1360.25, 155.36 ,153.54,
137.52, 246.17, 947.04)
>
>
> # Perform partial correlation analysis
> results1 <- parcor_ijk(R, E, M)

> results2 <- parcor_ijk(R, E, L)

> results3 <- parcor_ijk(R, M, E)

> results4 <- parcor_ijk(R, M, L)

> results5 <- parcor_ijk(R, L, M)

> results6 <- parcor_ijk(R, L, E)

>
> # Function to calculate p-values for partial correlations
> calculate_p_value <- function(partial_correlation, n) {
+ t_value <- partial_correlation * sqrt((n - 3) / (1 - partial_correlation^2))
+ p_value <- 2 * pt(-abs(t_value), df = n - 2)
+ return(p_value)
+ }
>
> partial_correlations1 <- as.numeric(results1["ouij"])
> partial_correlations2 <- as.numeric(results2["ouij"])
```

```

> partial_correlations3 <- as.numeric(results3["ouij"])
> partial_correlations4 <- as.numeric(results4["ouij"])
> partial_correlations5 <- as.numeric(results5["ouij"])
> partial_correlations6 <- as.numeric(results6["ouij"])
>
> # Number of observations (replace with your actual number of observations)
> n <- 16
> # Calculate p-values
> p_values1 <- sapply(partial_correlations1, calculate_p_value, n = n)
> p_values2 <- sapply(partial_correlations2, calculate_p_value, n = n)
> p_values3 <- sapply(partial_correlations3, calculate_p_value, n = n)
> p_values4 <- sapply(partial_correlations4, calculate_p_value, n = n)
> p_values5 <- sapply(partial_correlations5, calculate_p_value, n = n)
> p_values6 <- sapply(partial_correlations6, calculate_p_value, n = n)
> # Display results
> results_with_p_values1 <- cbind(results1, p_values = p_values1)
> results_with_p_values2 <- cbind(results2, p_values = p_values2)
> results_with_p_values3 <- cbind(results3, p_values = p_values3)
> results_with_p_values4 <- cbind(results4, p_values = p_values4)
> results_with_p_values5 <- cbind(results5, p_values = p_values5)
> results_with_p_values6 <- cbind(results6, p_values = p_values6)
>
> print(results_with_p_values1)
      results1  p_values
ouij -0.675627 0.00521839
ouji -0.457215 0.00521839
> print(results_with_p_values2)
      results2  p_values
ouij 0.3578286 0.1887304
ouji 0.0002272097 0.1887304
> print(results_with_p_values3)
      results3  p_values
ouij -0.002204927 0.9937691
ouji -0.1607653 0.9937691
>
> print(results_with_p_values4)
      results4  p_values
ouij 0.1302804 0.6429669
ouji 0.2366568 0.6429669
> print(results_with_p_values5)
      results5  p_values
ouij -0.3868275 0.152657
ouji -0.277596 0.152657
> print(results_with_p_values6)
      results6  p_values
ouij 0.02199687 0.9378929
ouji 0.0003193682 0.9378929
> # For dependence
> cor(R,E)
[1] -0.2198861
> depMeas(R, E, blksiz = length(R))
[1] -0.8915622
> depMeas(E, R, blksiz = length(E))
[1] -0.8915622
>
>
> gmcxy_np(R,E)
$corxy
[1] 0.7922567

$coryx
[1] 0.3355778

> gmcxy_np(E,R)
$corxy
[1] 0.3355778

$coryx
[1] 0.7922567

>
>
>
>

```