

ANALISIS DATA

(INSTRUKSI DAN HASIL ANALISIS R STUDIO)

I. INSTRUKSI UMUM R STUDIO

1. PERSIAPAN LINGKUNGAN

```
install.packages(c("readxl","dplyr","tidyr","psych","GPArotation"))
```

```
library(readxl)
```

```
library(dplyr)
```

```
library(tidyr)
```

```
library(psych)
```

```
library(GPArotation)
```

2. IMPORT DATA

3. CEK DATA

```
raw <- read_excel("nama file.xlsx")
```

```
ID <- raw[[1]]
```

```
resp <- raw[, -1]
```

4. DEFINE PAIR KEY

```
pair_key <- data.frame(  
  item = paste0("Q", 1:30),  
  left = c("I","A","R","S","R","I","S","A","R","I",  
    "R","E","A","I","R","A","R","E","I","S",  
    "R","I","R","I","A","R","A","S","I","R"),  
  right = c("S","E","I","E","A","C","C","C","S","A",  
    "C","C","S","E","E","S","A","C","E","C",  
    "S","A","C","S","C","E","E","E","C","I"),  
  stringsAsFactors = FALSE  
)
```

5. KONVERSI JAWABAN

```
score_pair <- function(answer, left, right) {  
  ifelse(answer == left, 1,  
    ifelse(answer == right, 0, NA))  
}
```

Generalisir ke semua nomor

```
for (i in 1:nrow(pair_key)) {  
  q <- pair_key$item[i]  
  lp <- pair_key$left[i]  
  rp <- pair_key$right[i]  
  
  resp[[paste0(q, "_p")]] <- score_pair(resp[[q]], lp, rp)  
}
```

6. MAPPING AITEM

```
cols_R <- c("Q3_p", "Q5_p", "Q9_p", "Q11_p", "Q15_p",  
           "Q17_p", "Q21_p", "Q23_p", "Q26_p", "Q30_p")  
  
cols_I <- c("Q1_p", "Q6_p", "Q10_p", "Q14_p", "Q19_p",  
           "Q22_p", "Q24_p", "Q29_p", "Q2_p", "Q12_p")  
  
cols_A <- c("Q2_p", "Q8_p", "Q13_p", "Q16_p", "Q17_p",  
           "Q22_p", "Q25_p", "Q27_p", "Q5_p", "Q3_p")  
  
cols_S <- c("Q1_p", "Q4_p", "Q7_p", "Q9_p", "Q21_p",  
           "Q24_p", "Q28_p", "Q16_p", "Q13_p", "Q20_p")  
  
cols_E <- c("Q2_p", "Q4_p", "Q12_p", "Q14_p", "Q15_p",  
           "Q18_p", "Q19_p", "Q26_p", "Q27_p", "Q28_p")  
  
cols_C <- c("Q6_p", "Q7_p", "Q8_p", "Q11_p", "Q12_p",  
           "Q18_p", "Q20_p", "Q23_p", "Q25_p", "Q29_p")
```

7. HITUNG SKOR DIMENSI

```
score_dim <- data.frame(  
  ID = ID,  
  R = rowMeans(resp[cols_R], na.rm = TRUE),  
  I = rowMeans(resp[cols_I], na.rm = TRUE),  
  A = rowMeans(resp[cols_A], na.rm = TRUE),  
  S = rowMeans(resp[cols_S], na.rm = TRUE),  
  E = rowMeans(resp[cols_E], na.rm = TRUE),  
  C = rowMeans(resp[cols_C], na.rm = TRUE)  
)
```

8. HITUNG OMEGA

```
omega_R <- omega(resp[cols_R], nfactors = 1)  
omega_I <- omega(resp[cols_I], nfactors = 1)  
omega_A <- omega(resp[cols_A], nfactors = 1)  
omega_S <- omega(resp[cols_S], nfactors = 1)
```

```
omega_E <- omega(resp[cols_E], nfactors = 1)
omega_C <- omega(resp[cols_C], nfactors = 1)
```

9. CEK AUDIT DATA

```
pair_long <- pair_key %>%
  pivot_longer(cols = c(left, right), values_to = "domain")

table(pair_long$domain)
```

10.SIMPAN NILAI OMEGA

```
omega_values <- data.frame(
  Domain = c("R", "I", "A", "S", "E", "C"),
  Omega_T = c(
    omega_R$omega.tot,
    omega_I$omega.tot,
    omega_A$omega.tot,
    omega_S$omega.tot,
    omega_E$omega.tot,
    omega_C$omega.tot
  )
)

omega_values
```

11.TAMBAHKAN JUMLAH AITEM PER DOMAIN

```
omega_values$N_item <- c(
  length(cols_R),
  length(cols_I),
  length(cols_A),
  length(cols_S),
  length(cols_E),
  length(cols_C)
)

omega_values
```

12.UBAH FORMAT UNTUK MEMBUAT MATRIX

```
library(dplyr)
library(tidyr)

pair_long <- data.frame(
  ID = rep(ID, times = nrow(pair_key)),
```

```

item = rep(pair_key$item, each = length(ID)),
left = rep(pair_key$left, each = length(ID)),
right = rep(pair_key$right, each = length(ID)),
score = unlist(resp[paste0(pair_key$item, "_p")])
)

```

13.MENENTUKAN PEMENANG

```

pair_long <- pair_long %>%
  mutate(
    winner = ifelse(score == 1, left,
                    ifelse(score == 0, right, NA))
  )

```

14.MEMBUAT PAIR WINNING MATRIX

```

domains <- c("R", "I", "A", "S", "E", "C")

win_matrix <- matrix(0,
  nrow = 6,
  ncol = 6,
  dimnames = list(domains, domains))

for (i in 1:nrow(pair_long)) {
  w <- pair_long$winner[i]
  l <- ifelse(pair_long$left[i] == w,
    pair_long$right[i],
    pair_long$left[i])

  if (!is.na(w)) {
    win_matrix[w, l] <- win_matrix[w, l] + 1
  }
}

win_matrix

```

15.PROPORTIONAL PAIR WINNING MATRIX

```

prop_matrix <- matrix(NA,
  nrow = 6,
  ncol = 6,
  dimnames = list(domains, domains))

for (i in domains) {
  for (j in domains) {
    if (i != j) {
      wij <- win_matrix[i, j]
      wji <- win_matrix[j, i]
    }
  }
}

```

```

      prop_matrix[i, j] <- wij / (wij + wji)
    }
  }
}

```

```

round(prop_matrix, 3)

```

16.MEMBUAT THURSTONE SCALE VALUE

```

z_matrix <- qnorm(prop_matrix)

```

17.NILAI TSV PER DOMAIN

```

thurstone_scale <- rowMeans(z_matrix, na.rm = TRUE)

```

```

thurstone_scale

```

18.MENGHITUNG CONGRUEN INDEX

A. BUAT DISTANCE MATRIX MENGGUNAKAN TSV

```

D <- as.matrix(abs(outer(thurstone_scale,
      thurstone_scale,
      "-")))

```

B. UBAH DISTANCE MENJADI SIMILARITY

```

D_scaled <- D / max(D)

```

```

R_emp <- 1 - D_scaled

```

```

diag(R_emp) <- 1

```

```

round(R_emp, 3)

```

C. UBAH MENJADI MATRIX KORELASI EMPIRIS (UBAH ANGKA SESUAI EMPIRIS SESUAI DI B)

```

R_emp <- matrix(c(
  1.00, 0.79, 0.55, 0.64, 0.61, 0.94,
  0.79, 1.00, 0.66, 0.59, 0.70, 0.79,
  0.55, 0.66, 1.00, 0.63, 0.64, 0.69,
  0.64, 0.59, 0.63, 1.00, 0.62, 0.85,
  0.61, 0.70, 0.64, 0.62, 1.00, 0.91,
  0.94, 0.79, 0.69, 0.85, 0.91, 1.00
), nrow = 6, byrow = TRUE)

```

```

colnames(R_emp) <- rownames(R_emp) <- c("R","I","A","S","E","C")

```

D. BANDINGKAN DENGAN MATRIX TEORI

```

R_ideal <- matrix(c(
  1, 1, .5, 0, .5, 1,
  1, 1, 1, .5, 0, .5,
  .5, 1, 1, 1, .5, 0,
  0, .5, 1, 1, 1, .5,
  .5, 0, .5, 1, 1, 1,
  1, .5, 0, .5, 1, 1
), nrow = 6, byrow = TRUE)

colnames(R_ideal) <- rownames(R_ideal) <- c("R", "I", "A", "S", "E", "C")

```

E. HITUNG CONGRUEN INDEX

```

upper_emp <- R_emp[upper.tri(R_emp)]
upper_ideal <- R_ideal[upper.tri(R_ideal)]

CI <- cor(upper_emp, upper_ideal)
CI

```

19. PLOT EMPIRIS

A. PANGGIL TSV

```
thurstone_scale
```

B. BUAT DISTANCE MATRIX

```

D <- as.matrix(abs(outer(thurstone_scale,
  thurstone_scale,
  "-")))

```

C. UBAH KE BENTUK DIST

```
dist_emp <- as.dist(D)
```

D. JALANKAN MDS

```
mds <- cmdscale(dist_emp, k = 2, eig = TRUE)
```

E. BENTUK KONFIGURASI STRUKTUR

```
score_dim <- as.data.frame(mds$points)
```

```
colnames(score_dim) <- c("Dim1", "Dim2")
```

```
score_dim$Domain <- names(thurstone_scale)
```

```
score_dim
```

F. KONFIGURASI HEXAGONAL

```

plot(score_dim$Dim1,
     score_dim$Dim2,
     type="n",
     xlab="Dim1",
     ylab="Dim2",
     main="MDS Configuration (Empirical RIASEC)")

text(score_dim$Dim1,
     score_dim$Dim2,
     labels=score_dim$Domain,
     cex=1.2)

```

20. PLOT IDEAL DAN EMPIRIS

A. IDEAL

```

theta <- seq(0, 2*pi, length.out = 7)[-7]

ideal_coords <- data.frame(
  x = cos(theta),
  y = sin(theta),
  Domain = c("R", "I", "A", "S", "E", "C")
)

```

B. EMPIRIS (isi sesuai dengan data empiris)

```

emp_coords <- data.frame(
  x = c(0.35, 0.25, -0.10, 0.15, 0.28, -0.55),
  y = c(0.25, 0.05, 0.22, -0.18, 0.02, 0.12),
  Domain = c("R", "I", "A", "S", "E", "C")
)

```

C. BERDAMPINGAN

```

par(mfrow = c(1,2))

# Ideal
plot(ideal_coords$x, ideal_coords$y, type="n",
     main="Hexagonal Ideal Holland",
     xlab="Dimensi 1", ylab="Dimensi 2")
text(ideal_coords$x, ideal_coords$y, ideal_coords$Domain)
lines(ideal_coords$x[c(1:6,1)], ideal_coords$y[c(1:6,1)])

# Empiris
plot(emp_coords$x, emp_coords$y, type="n",
     main="Hexagonal RIASEC Empiris",
     xlab="Dimensi 1", ylab="Dimensi 2")
text(emp_coords$x, emp_coords$y, emp_coords$Domain)

```

```
lines(emp_coords$x[c(1:6,1)], emp_coords$y[c(1:6,1)])
```

21.MENANDAI PELANGGARAN

A. URUTAN TEORETIS

```
hex_order <- c("R","I","A","S","E","C")
```

B. MENGHITUNG JARAK TEORETIS

```
hex_distance <- function(d1, d2, order) {  
  i1 <- match(d1, order)  
  i2 <- match(d2, order)  
  raw <- abs(i1 - i2)  
  min(raw, length(order) - raw)  
}
```

C. JARAK EMPIRIS

```
cor_mat <- cor(score_dim[, -1], use = "pairwise.complete.obs")  
dist_emp <- as.matrix(1 - cor_mat)
```

D. TABEL VARIASI PELANGGARAN

```
library(dplyr)
```

```
domains <- colnames(dist_emp)
```

```
violation_table <- expand.grid(  
  d1 = domains,  
  d2 = domains,  
  stringsAsFactors = FALSE  
) %>%
```

```
filter(d1 < d2) %>%
```

```
mutate(  
  hex_step = mapply(hex_distance, d1, d2,  
    MoreArgs = list(order = hex_order)),  
  expected_dist = case_when(  
    hex_step == 1 ~ "close",  
    hex_step == 2 ~ "moderate",
```

```

    hex_step == 3 ~ "far"
  ),
  empirical_dist = dist_emp[cbind(match(d1, domains),
                                   match(d2, domains))],
  empirical_cat = case_when(
    empirical_dist <= 0.30 ~ "close",
    empirical_dist <= 0.50 ~ "moderate",
    TRUE ~ "far"
  ),
  violation = expected_dist != empirical_cat
)

```

II. HASIL ANALISIS DATA

1. DATA POPULASI

R version 4.5.2 (2025-10-31 ucrt) -- "[Not] Part in a Rumble"
 Copyright (C) 2025 The R Foundation for Statistical Computing
 Platform: x86_64-w64-mingw32/x64

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.

```
> install.packages(c("readxl","dplyr","tidyr","psych","GPArotation"))
```

WARNING: Rtools is required to build R packages but is not currently installed. Please download and install the appropriate version of Rtools before proceeding:

<https://cran.rstudio.com/bin/windows/Rtools/>

installing packages into 'C:/Users/MyPC One Pro L/AppData/Local/R/win-library/4.5'

(as 'lib' is unspecified)trying URL

'https://cran.rstudio.com/bin/windows/contrib/4.5/readxl_1.4.5.zip'

trying URL

'https://cran.rstudio.com/bin/windows/contrib/4.5/dplyr_1.2.0.zip'

trying URL

'https://cran.rstudio.com/bin/windows/contrib/4.5/tidyr_1.3.2.zip'

trying URL

'https://cran.rstudio.com/bin/windows/contrib/4.5/psych_2.6.1.zip'

trying URL

'https://cran.rstudio.com/bin/windows/contrib/4.5/GPARotation_2025.3-1.zip'

package 'readxl' successfully unpacked and MD5 sums checked

package 'dplyr' successfully unpacked and MD5 sums checked

package 'tidyr' successfully unpacked and MD5 sums checked

package 'psych' successfully unpacked and MD5 sums checked

package 'GPARotation' successfully unpacked and MD5 sums checked

The downloaded binary packages are in

C:\Users\MyPC One Pro

L\AppData\Local\Temp\RtmpKOYU0q\downloaded_packages

> library(readxl)

> library(dplyr)

Attaching package: 'dplyr'

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

filter, lag

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

intersect, setdiff, setequal, union> library(tidyr)

> library(psych)

> library(GPARotation)

Attaching package: 'GPARotation'

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

equamax, varimin>

> library(readxl)

> data_RIASEC <- read_excel("data_RIASEC.xlsx")

> View(data_RIASEC)

> raw <- read_excel("data_RIASEC.xlsx")

>

```

> ID <- raw[[1]]
> resp <- raw[, -1]
>
> pair_key <- data.frame(
+   item = paste0("Q", 1:30),
+   left = c("I","A","R","S","R","I","S","A","R","I",
+            "R","E","A","I","R","A","R","E","I","S",
+            "R","I","R","I","A","R","A","S","I","R"),
+   right = c("S","E","I","E","A","C","C","C","S","A",
+            "C","C","S","E","E","S","A","C","E","C",
+            "S","A","C","S","C","E","E","E","C","I"),
+   stringsAsFactors = FALSE
+ )
>
> score_pair <- function(answer, left, right) {
+   ifelse(answer == left, 1,
+         ifelse(answer == right, 0, NA))
+ }
>
> for (i in 1:nrow(pair_key)) {
+   q <- pair_key$item[i]
+   lp <- pair_key$left[i]
+   rp <- pair_key$right[i]
+
+   resp[[paste0(q, "_p")]] <- score_pair(resp[[q]], lp, rp)
+ }
>
> cols_R <- c("Q3_p", "Q5_p", "Q9_p", "Q11_p", "Q15_p",
+            "Q17_p", "Q21_p", "Q23_p", "Q26_p", "Q30_p")
>
> cols_I <- c("Q1_p", "Q6_p", "Q10_p", "Q14_p", "Q19_p",
+            "Q22_p", "Q24_p", "Q29_p", "Q2_p", "Q12_p")
>
> cols_A <- c("Q2_p", "Q8_p", "Q13_p", "Q16_p", "Q17_p",
+            "Q22_p", "Q25_p", "Q27_p", "Q5_p", "Q3_p")
>
> cols_S <- c("Q1_p", "Q4_p", "Q7_p", "Q9_p", "Q21_p",
+            "Q24_p", "Q28_p", "Q16_p", "Q13_p", "Q20_p")
>
> cols_E <- c("Q2_p", "Q4_p", "Q12_p", "Q14_p", "Q15_p",
+            "Q18_p", "Q19_p", "Q26_p", "Q27_p", "Q28_p")
>
> cols_C <- c("Q6_p", "Q7_p", "Q8_p", "Q11_p", "Q12_p",
+            "Q18_p", "Q20_p", "Q23_p", "Q25_p", "Q29_p")
>
> score_dim <- data.frame(
+   ID = ID,
+   R = rowMeans(resp[cols_R], na.rm = TRUE),
+   I = rowMeans(resp[cols_I], na.rm = TRUE),
+   A = rowMeans(resp[cols_A], na.rm = TRUE),
+   S = rowMeans(resp[cols_S], na.rm = TRUE),
+   E = rowMeans(resp[cols_E], na.rm = TRUE),
+   C = rowMeans(resp[cols_C], na.rm = TRUE)

```

```

+ )
>
> omega_R <- omega(resp[cols_R], nfactors = 1)
Omega_h for 1 factor is not meaningful, just omega_t
Warning message:
In schmid(m, nfactors, fm, digits, rotate = rotate, n.obs = n.obs, :
  Omega_h and Omega_asymptotic are not meaningful with one factor>
omega_I <- omega(resp[cols_I], nfactors = 1)
Omega_h for 1 factor is not meaningful, just omega_t
Warning message:
In schmid(m, nfactors, fm, digits, rotate = rotate, n.obs = n.obs, :
  Omega_h and Omega_asymptotic are not meaningful with one factor>
omega_A <- omega(resp[cols_A], nfactors = 1)
Omega_h for 1 factor is not meaningful, just omega_t
Warning message:
In schmid(m, nfactors, fm, digits, rotate = rotate, n.obs = n.obs, :
  Omega_h and Omega_asymptotic are not meaningful with one factor>
omega_S <- omega(resp[cols_S], nfactors = 1)
Omega_h for 1 factor is not meaningful, just omega_t
Warning message:
In schmid(m, nfactors, fm, digits, rotate = rotate, n.obs = n.obs, :
  Omega_h and Omega_asymptotic are not meaningful with one factor>
omega_E <- omega(resp[cols_E], nfactors = 1)
Omega_h for 1 factor is not meaningful, just omega_t
Warning message:
In schmid(m, nfactors, fm, digits, rotate = rotate, n.obs = n.obs, :
  Omega_h and Omega_asymptotic are not meaningful with one factor>
omega_C <- omega(resp[cols_C], nfactors = 1)
Omega_h for 1 factor is not meaningful, just omega_t
Warning message:
In schmid(m, nfactors, fm, digits, rotate = rotate, n.obs = n.obs, :
  Omega_h and Omega_asymptotic are not meaningful with one factor>
> pair_long <- pair_key %>%
+   pivot_longer(cols = c(left, right), values_to = "domain")
>
> table(pair_long$domain)

  A  C  E  I  R  S
10 10 10 10 10 10
>
> omega_values <- data.frame(
+   Domain = c("R", "I", "A", "S", "E", "C"),
+   Omega_T = c(
+     omega_R$omega.tot,

```

```

+         omega_I$omega.tot,
+         omega_A$omega.tot,
+         omega_S$omega.tot,
+         omega_E$omega.tot,
+         omega_C$omega.tot
+     )
+ )
>
> omega_values
  Domain   Omega_T
1      R 0.4552188
2      I 0.4737504
3      A 0.4171395
4      S 0.5953714
5      E 0.6051682
6      C 0.6738918
>
> omega_values$N_item <- c(
+   length(cols_R),
+   length(cols_I),
+   length(cols_A),
+   length(cols_S),
+   length(cols_E),
+   length(cols_C)
+ )
>
> omega_values
  Domain   Omega_T N_item
1      R 0.4552188    10
2      I 0.4737504    10
3      A 0.4171395    10
4      S 0.5953714    10
5      E 0.6051682    10
6      C 0.6738918    10
>
> library(dplyr)
> library(tidyr)
>
> pair_long <- data.frame(
+   ID = rep(ID, times = nrow(pair_key)),
+   item = rep(pair_key$item, each = length(ID)),
+   left = rep(pair_key$left, each = length(ID)),
+   right = rep(pair_key$right, each = length(ID)),
+   score = unlist(resp[paste0(pair_key$item, "_p")])
+ )
>
> pair_long <- pair_long %>%
+   mutate(
+     winner = ifelse(score == 1, left,
+                     ifelse(score == 0, right, NA))
+   )
>
> domains <- c("R","I","A","S","E","C")

```

```

>
> win_matrix <- matrix(0,
+                       nrow = 6,
+                       ncol = 6,
+                       dimnames = list(domains, domains))
>
> for (i in 1:nrow(pair_long)) {
+   w <- pair_long$winner[i]
+   l <- ifelse(pair_long$left[i] == w,
+               pair_long$right[i],
+               pair_long$left[i])
+
+   if (!is.na(w)) {
+     win_matrix[w, l] <- win_matrix[w, l] + 1
+   }
+ }
>
> win_matrix
      R      I      A      S      E      C
R      0 3279 6658 5026 5998 2464
I 6219      0 6665 3293 7893 3992
A 2840 2833      0 6471 5113 4565
S 4472 6205 3027      0 5644 4598
E 3500 1605 4385 3854      0 4210
C 7034 5506 4933 4900 5288      0
>
> prop_matrix <- matrix(NA,
+                       nrow = 6,
+                       ncol = 6,
+                       dimnames = list(domains, domains))
>
> for (i in domains) {
+   for (j in domains) {
+     if (i != j) {
+       wij <- win_matrix[i, j]
+       wji <- win_matrix[j, i]
+       prop_matrix[i, j] <- wij / (wij + wji)
+     }
+   }
+ }
>
> round(prop_matrix, 3)
      R      I      A      S      E      C
R      NA 0.345 0.701 0.529 0.632 0.259
I 0.655      NA 0.702 0.347 0.831 0.420
A 0.299 0.298      NA 0.681 0.538 0.481
S 0.471 0.653 0.319      NA 0.594 0.484
E 0.368 0.169 0.462 0.406      NA 0.443
C 0.741 0.580 0.519 0.516 0.557      NA
>
> z_matrix <- qnorm(prop_matrix)
>
> thurstone_scale <- rowMeans(z_matrix, na.rm = TRUE)

```

```

>
> thurstone_scale
      R      I      A      S
-0.021420882 0.258086626 -0.107529640 0.009660593
      E      C
-0.354281684 0.215484986
>

> thurstone_scale
      R      I      A      S
-0.021420882 0.258086626 -0.107529640 0.009660593
      E      C
-0.354281684 0.215484986
>
> R_emp <- matrix(c(
+   1.00, 0.21, 0.44, 0.36, 0.39, 0.06,
+   0.21, 1.00, 0.34, 0.41, 0.30, 0.21,
+   0.44, 0.34, 1.00, 0.37, 0.36, 0.31,
+   0.36, 0.41, 0.37, 1.00, 0.38, 0.15,
+   0.39, 0.30, 0.36, 0.38, 1.00, 0.09,
+   0.06, 0.21, 0.31, 0.15, 0.09, 1.00
+ ), nrow = 6, byrow = TRUE)
>
> colnames(R_emp) <- rownames(R_emp) <- c("R","I","A","S","E","C")
>
> R_ideal <- matrix(c(
+   1, 1, .5, 0, .5, 1,
+   1, 1, 1, .5, 0, .5,
+   .5, 1, 1, 1, .5, 0,
+   0, .5, 1, 1, 1, .5,
+   .5, 0, .5, 1, 1, 1,
+   1, .5, 0, .5, 1, 1
+ ), nrow = 6, byrow = TRUE)
>
> colnames(R_ideal) <- rownames(R_ideal) <-
c("R","I","A","S","E","C")
>
> upper_emp <- R_emp[upper.tri(R_emp)]
> upper_ideal <- R_ideal[upper.tri(R_ideal)]
>
> CI <- cor(upper_emp, upper_ideal)
> CI
[1] -0.3059493
>
> theta <- seq(0, 2*pi, length.out = 7)[-7]
>
> ideal_coords <- data.frame(
+   x = cos(theta),
+   y = sin(theta),
+   Domain = c("R","I","A","S","E","C")
+ )
>
> emp_coords <- data.frame(

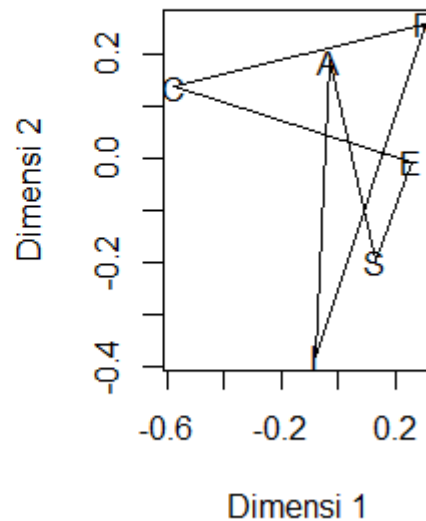
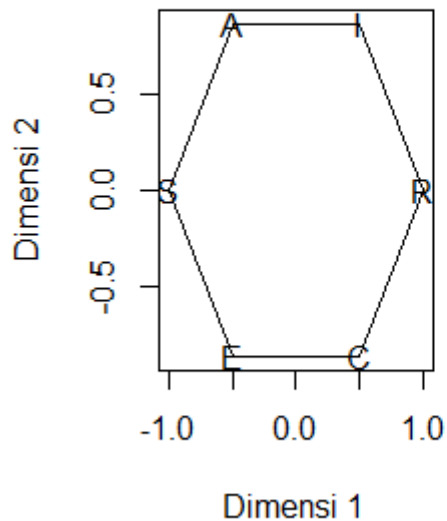
```

```

+ x = c(0.35, 0.25, -0.10, 0.15, 0.28, -0.55),
+ y = c(0.25, 0.05, 0.22, -0.18, 0.02, 0.12),
+ Domain = c("R","I","A","S","E","C")
+ )
>
>

```

Hexagonal Ideal Holland Hexagonal RIASEC Empir



2. DATA JABATAN FUNGSIONAL GURU

R version 4.5.2 (2025-10-31 ucrt) -- "[Not] Part in a Rumble"
 Copyright (C) 2025 The R Foundation for Statistical Computing
 Platform: x86_64-w64-mingw32/x64

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.

```
> install.packages(c("readxl","dplyr","tidyr","psych","GPARotation"))
```

WARNING: Rtools is required to build R packages but is not currently installed. Please download and install the appropriate version of Rtools before proceeding:

<https://cran.rstudio.com/bin/windows/Rtools/>

installing packages into 'C:/Users/MyPC One Pro L/AppData/Local/R/win-library/4.5'

(as 'lib' is unspecified)trying URL

'https://cran.rstudio.com/bin/windows/contrib/4.5/readxl_1.4.5.zip'

trying URL

'https://cran.rstudio.com/bin/windows/contrib/4.5/dplyr_1.2.0.zip'

trying URL

'https://cran.rstudio.com/bin/windows/contrib/4.5/tidyr_1.3.2.zip'

trying URL

'https://cran.rstudio.com/bin/windows/contrib/4.5/psych_2.6.1.zip'

trying URL

'https://cran.rstudio.com/bin/windows/contrib/4.5/GPARotation_2025.3-1.zip'

package 'readxl' successfully unpacked and MD5 sums checked

package 'dplyr' successfully unpacked and MD5 sums checked

package 'tidyr' successfully unpacked and MD5 sums checked

package 'psych' successfully unpacked and MD5 sums checked

package 'GPARotation' successfully unpacked and MD5 sums checked

The downloaded binary packages are in

C:\Users\MyPC One Pro

L\AppData\Local\Temp\RtmpU9PDvN\downloaded_packages

```
> library(readxl)
```

```
> library(dplyr)
```

Attaching package: 'dplyr'

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

filter, lag

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

intersect, setdiff, setequal, union> library(tidyr)

```
> library(psych)
```

```
> library(GPARotation)
```

Attaching package: 'GPARotation'

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

```
equamax, varimin>
> library(readxl)
> guru <- read_excel("guru.xlsx")
> View(guru)
> raw <- read_excel("guru.xlsx")
>
> ID <- raw[[1]]
> resp <- raw[, -1]
>
> pair_key <- data.frame(
+   item = paste0("Q", 1:30),
+   left = c("I","A","R","S","R","I","S","A","R","I",
+            "R","E","A","I","R","A","R","E","I","S",
+            "R","I","R","I","A","R","A","S","I","R"),
+   right = c("S","E","I","E","A","C","C","C","S","A",
+            "C","C","S","E","E","S","A","C","E","C",
+            "S","A","C","S","C","E","E","E","C","I"),
+   stringsAsFactors = FALSE
+ )
>
> score_pair <- function(answer, left, right) {
+   ifelse(answer == left, 1,
+         ifelse(answer == right, 0, NA))
+ }
>
> for (i in 1:nrow(pair_key)) {
+   q <- pair_key$item[i]
+   lp <- pair_key$left[i]
+   rp <- pair_key$right[i]
+
+   resp[[paste0(q, "_p")]] <- score_pair(resp[[q]], lp, rp)
+ }
>
> cols_R <- c("Q3_p","Q5_p","Q9_p","Q11_p","Q15_p",
+            "Q17_p","Q21_p","Q23_p","Q26_p","Q30_p")
>
> cols_I <- c("Q1_p","Q6_p","Q10_p","Q14_p","Q19_p",
+            "Q22_p","Q24_p","Q29_p","Q2_p","Q12_p")
>
> cols_A <- c("Q2_p","Q8_p","Q13_p","Q16_p","Q17_p",
+            "Q22_p","Q25_p","Q27_p","Q5_p","Q3_p")
>
> cols_S <- c("Q1_p","Q4_p","Q7_p","Q9_p","Q21_p",
+            "Q24_p","Q28_p","Q16_p","Q13_p","Q20_p")
>
> cols_E <- c("Q2_p","Q4_p","Q12_p","Q14_p","Q15_p",
+            "Q18_p","Q19_p","Q26_p","Q27_p","Q28_p")
>
> cols_C <- c("Q6_p","Q7_p","Q8_p","Q11_p","Q12_p",
+            "Q18_p","Q20_p","Q23_p","Q25_p","Q29_p")
```

```

>
> score_dim <- data.frame(
+   ID = ID,
+   R = rowMeans(resp[cols_R], na.rm = TRUE),
+   I = rowMeans(resp[cols_I], na.rm = TRUE),
+   A = rowMeans(resp[cols_A], na.rm = TRUE),
+   S = rowMeans(resp[cols_S], na.rm = TRUE),
+   E = rowMeans(resp[cols_E], na.rm = TRUE),
+   C = rowMeans(resp[cols_C], na.rm = TRUE)
+ )
>
> omega_R <- omega(resp[cols_R], nfactors = 1)
Omega_h for 1 factor is not meaningful, just omega_t
Warning message:
In schmid(m, nfactors, fm, digits, rotate = rotate, n.obs = n.obs, :
  Omega_h and Omega_asymptotic are not meaningful with one factor>
omega_I <- omega(resp[cols_I], nfactors = 1)
Omega_h for 1 factor is not meaningful, just omega_t
Warning message:
In schmid(m, nfactors, fm, digits, rotate = rotate, n.obs = n.obs, :
  Omega_h and Omega_asymptotic are not meaningful with one factor>
omega_A <- omega(resp[cols_A], nfactors = 1)
Omega_h for 1 factor is not meaningful, just omega_t
Warning message:
In schmid(m, nfactors, fm, digits, rotate = rotate, n.obs = n.obs, :
  Omega_h and Omega_asymptotic are not meaningful with one factor>
omega_S <- omega(resp[cols_S], nfactors = 1)
Omega_h for 1 factor is not meaningful, just omega_t
Warning message:
In schmid(m, nfactors, fm, digits, rotate = rotate, n.obs = n.obs, :
  Omega_h and Omega_asymptotic are not meaningful with one factor>
omega_E <- omega(resp[cols_E], nfactors = 1)
Omega_h for 1 factor is not meaningful, just omega_t
Warning message:
In schmid(m, nfactors, fm, digits, rotate = rotate, n.obs = n.obs, :
  Omega_h and Omega_asymptotic are not meaningful with one factor>
omega_C <- omega(resp[cols_C], nfactors = 1)
Omega_h for 1 factor is not meaningful, just omega_t
Warning message:
In schmid(m, nfactors, fm, digits, rotate = rotate, n.obs = n.obs, :
  Omega_h and Omega_asymptotic are not meaningful with one factor>
> pair_long <- pair_key %>%
+   pivot_longer(cols = c(left, right), values_to = "domain")
>

```

```

> table(pair_long$domain)

  A  C  E  I  R  S 
10 10 10 10 10 10 
> 
> omega_values <- data.frame(
+   Domain = c("R","I","A","S","E","C"),
+   Omega_T = c(
+     omega_R$omega.tot,
+     omega_I$omega.tot,
+     omega_A$omega.tot,
+     omega_S$omega.tot,
+     omega_E$omega.tot,
+     omega_C$omega.tot
+   )
+ )
> 
> omega_values
  Domain  Omega_T
1      R 0.4252824
2      I 0.4251490
3      A 0.3999575
4      S 0.5827790
5      E 0.5742544
6      C 0.6141258
> 
> omega_values$N_item <- c(
+   length(cols_R),
+   length(cols_I),
+   length(cols_A),
+   length(cols_S),
+   length(cols_E),
+   length(cols_C)
+ )
> 
> omega_values
  Domain  Omega_T N_item
1      R 0.4252824     10
2      I 0.4251490     10
3      A 0.3999575     10
4      S 0.5827790     10
5      E 0.5742544     10
6      C 0.6141258     10
> 
> library(dplyr)
> library(tidyr)
> 
> pair_long <- data.frame(
+   ID = rep(ID, times = nrow(pair_key)),
+   item = rep(pair_key$item, each = length(ID)),
+   left = rep(pair_key$left, each = length(ID)),
+   right = rep(pair_key$right, each = length(ID)),
+   score = unlist(resp[paste0(pair_key$item, "_p")])

```

```

+ )
>
> pair_long <- pair_long %>%
+   mutate(
+     winner = ifelse(score == 1, left,
+                     ifelse(score == 0, right, NA))
+   )
>
> domains <- c("R","I","A","S","E","C")
>
> win_matrix <- matrix(0,
+                       nrow = 6,
+                       ncol = 6,
+                       dimnames = list(domains, domains))
>
> for (i in 1:nrow(pair_long)) {
+   w <- pair_long$winner[i]
+   l <- ifelse(pair_long$left[i] == w,
+               pair_long$right[i],
+               pair_long$left[i])
+
+   if (!is.na(w)) {
+     win_matrix[w, l] <- win_matrix[w, l] + 1
+   }
+ }
>
> win_matrix
      R      I      A      S      E      C
R      0 1958 3577 2909 3504 1753
I 3640      0 3567 1884 4699 2396
A 2021 2031      0 3890 3301 3006
S 2689 3714 1708      0 3478 2829
E 2094  899 2297 2120      0 2482
C 3845 3202 2592 2769 3116      0
>
> prop_matrix <- matrix(NA,
+                       nrow = 6,
+                       ncol = 6,
+                       dimnames = list(domains, domains))
>
> for (i in domains) {
+   for (j in domains) {
+     if (i != j) {
+       wij <- win_matrix[i, j]
+       wji <- win_matrix[j, i]
+       prop_matrix[i, j] <- wij / (wij + wji)
+     }
+   }
+ }
>
> round(prop_matrix, 3)
      R      I      A      S      E      C
R      NA 0.350 0.639 0.520 0.626 0.313

```

```

I 0.650    NA 0.637 0.337 0.839 0.428
A 0.361 0.363    NA 0.695 0.590 0.537
S 0.480 0.663 0.305    NA 0.621 0.505
E 0.374 0.161 0.410 0.379    NA 0.443
C 0.687 0.572 0.463 0.495 0.557    NA
>
> z_matrix <- qnorm(prop_matrix)
>
> thurstone_scale <- rowMeans(z_matrix, na.rm = TRUE)
>
> thurstone_scale
      R      I      A      S
-0.02935565 0.22511803 0.02452028 0.03703525
      E      C
-0.39822974 0.14091184
>
> dist_hex <- as.dist(1 - cor_mat)
Error: object 'cor_mat' not found

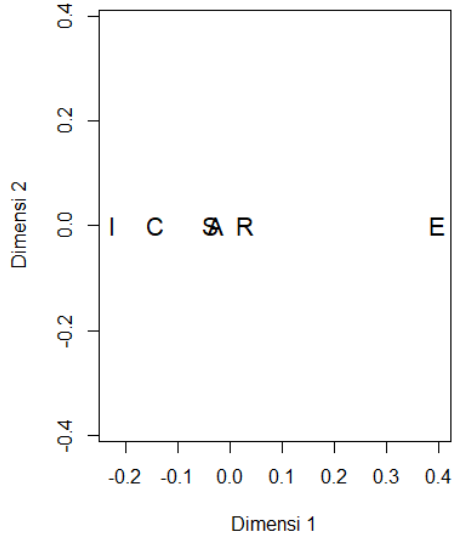
```

```

> thurstone_scale
      R      I      A      S
-0.02935565 0.22511803 0.02452028 0.03703525
      E      C
-0.39822974 0.14091184

```

Hexagonal RIASEC Empiris (Thurstone Case)



```

>
> D <- as.matrix(abs(outer(thurstone_scale,
+                           thurstone_scale,
+                           "-")))
>
> D_scaled <- D / max(D)
>
> R_emp <- 1 - D_scaled
>

```

```

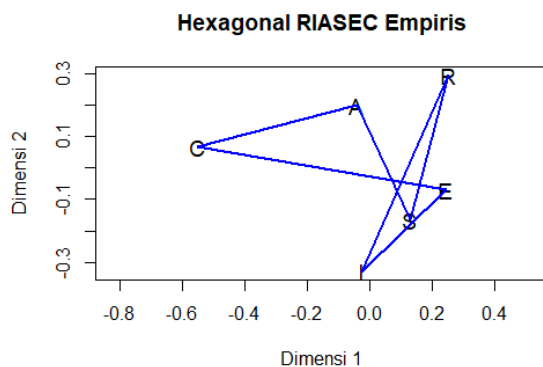
> diag(R_emp) <- 1
>
> round(R_emp, 3)
      R      I      A      S      E      C
R 1.000 0.592 0.914 0.893 0.408 0.727
I 0.592 1.000 0.678 0.698 0.000 0.865
A 0.914 0.678 1.000 0.980 0.322 0.813
S 0.893 0.698 0.980 1.000 0.302 0.833
E 0.408 0.000 0.322 0.302 1.000 0.135
C 0.727 0.865 0.813 0.833 0.135 1.000
>
> R_emp <- matrix(c(
+   1.00, 0.59, 0.92, 0.89, 0.41, 0.73,
+   0.59, 1.00, 0.68, 0.70, 0.00, 0.87,
+   0.91, 0.68, 1.00, 0.98, 0.32, 0.81,
+   0.89, 0.70, 0.98, 1.00, 0.30, 0.83,
+   0.41, 0.00, 0.32, 0.30, 1.00, 0.14,
+   0.73, 0.87, 0.81, 0.83, 0.14, 1.00
+ ), nrow = 6, byrow = TRUE)
>
> colnames(R_emp) <- rownames(R_emp) <- c("R","I","A","S","E","C")
>
> dist_emp <- as.dist(D)
>
> mds <- cmdscale(dist_emp, k=2, eig=TRUE)
>
> coords <- as.data.frame(mds$points)
> colnames(coords) <- c("Dim1","Dim2")
> coords$Domain <- names(thurstone_scale)
>
> coords
      Dim1      Dim2 Domain
R  0.02935565 0.000000e+00    R
I -0.22511803 -3.103693e-09    I
A -0.02452028 -1.695704e-09    A
S -0.03703525 -6.988373e-10    S
E  0.39822974 -1.650333e-09    E
C -0.14091184  7.731530e-10    C
>
> plot(coords$Dim1, coords$Dim2,
+       type="n",
+       asp=1,
+       xlab="Dimensi 1",
+       ylab="Dimensi 2",
+       main="Hexagonal RIASEC Empiris (Thurstone Case V)")
>
> text(coords$Dim1, coords$Dim2,
+       labels=coords$Domain,
+       cex=1.3)
>
> R_ideal <- matrix(c(
+   1, 1, .5, 0, .5, 1,
+   1, 1, 1, .5, 0, .5,

```

```

+      .5, 1, 1, 1, .5, 0,
+      0, .5, 1, 1, 1, .5,
+      .5, 0, .5, 1, 1, 1,
+      1, .5, 0, .5, 1, 1
+ ), nrow = 6, byrow = TRUE)
>
> colnames(R_ideal) <- rownames(R_ideal) <-
c("R","I","A","S","E","C")
>
> upper_emp <- R_emp[upper.tri(R_emp)]
> upper_ideal <- R_ideal[upper.tri(R_ideal)]
>
> CI <- cor(upper_emp, upper_ideal)
> CI
[1] -0.0343874
>
> theta <- seq(0, 2*pi, length.out = 7)[-7]
>
> ideal_coords <- data.frame(
+   x = cos(theta),
+   y = sin(theta),
+   Domain = c("R","I","A","S","E","C")
+ )
>
> emp_coords <- data.frame(
+   x = c(0.35, 0.25, -0.10, 0.15, 0.28, -0.55),
+   y = c(0.25, 0.05, 0.22, -0.18, 0.02, 0.12),
+   Domain = c("R","I","A","S","E","C")
+ )
>
> par(mfrow = c(1,2))

```



>

3. DATA JABATAN FUNGSIONAL NON GURU

R version 4.5.2 (2025-10-31 ucrt) -- "[Not] Part in a Rumble"
 Copyright (C) 2025 The R Foundation for Statistical Computing
 Platform: x86_64-w64-mingw32/x64

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.

```
> install.packages(c("readxl","dplyr","tidyr","psych","GPARotation"))
```

WARNING: Rtools is required to build R packages but is not currently installed. Please download and install the appropriate version of Rtools before proceeding:

<https://cran.rstudio.com/bin/windows/Rtools/>

Installing packages into 'C:/Users/MyPC One Pro L/AppData/Local/R/win-library/4.5'

```
(as 'lib' is unspecified)trying URL
'https://cran.rstudio.com/bin/windows/contrib/4.5/readxl_1.4.5.zip'
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.5/dplyr_1.2.0.zip'
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.5/tidyr_1.3.2.zip'
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.5/psych_2.6.1.zip'
trying URL
'https://cran.rstudio.com/bin/windows/contrib/4.5/GPARotation_2025.3-1.zip'
package 'readxl' successfully unpacked and MD5 sums checked
package 'dplyr' successfully unpacked and MD5 sums checked
package 'tidyr' successfully unpacked and MD5 sums checked
package 'psych' successfully unpacked and MD5 sums checked
package 'GPARotation' successfully unpacked and MD5 sums checked
```

The downloaded binary packages are in
C:\Users\MyPC One Pro
L\AppData\Local\Temp\RtmpQf32g0\downloaded_packages

```
> library(readxl)
> library(dplyr)
```

Attaching package: 'dplyr'

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

filter, lag

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

```

intersect, setdiff, setequal, union> library(tidyr)
> library(psych)
> library(GPARotation)

```

Attaching package: 'GPARotation'

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

```

equamax, varimin>
> library(readxl)
> jft <- read_excel("jft.xlsx")
> View(jft)
> raw <- read_excel("jft.xlsx")
>
> ID <- raw[[1]]
> resp <- raw[, -1]
>
> pair_key <- data.frame(
+   item = paste0("Q", 1:30),
+   left = c("I","A","R","S","R","I","S","A","R","I",
+            "R","E","A","I","R","A","R","E","I","S",
+            "R","I","R","I","A","R","A","S","I","R"),
+   right = c("S","E","I","E","A","C","C","C","S","A",
+            "C","C","S","E","E","S","A","C","E","C",
+            "S","A","C","S","C","E","E","E","C","I"),
+   stringsAsFactors = FALSE
+ )
>
> score_pair <- function(answer, left, right) {
+   ifelse(answer == left, 1,
+         ifelse(answer == right, 0, NA))
+ }
>
> for (i in 1:nrow(pair_key)) {
+   q <- pair_key$item[i]
+   lp <- pair_key$left[i]
+   rp <- pair_key$right[i]
+
+   resp[[paste0(q, "_p")]] <- score_pair(resp[[q]], lp, rp)
+ }
>
> cols_R <- c("Q3_p","Q5_p","Q9_p","Q11_p","Q15_p",
+            "Q17_p","Q21_p","Q23_p","Q26_p","Q30_p")
>
> cols_I <- c("Q1_p","Q6_p","Q10_p","Q14_p","Q19_p",
+            "Q22_p","Q24_p","Q29_p","Q2_p","Q12_p")
>
> cols_A <- c("Q2_p","Q8_p","Q13_p","Q16_p","Q17_p",
+            "Q22_p","Q25_p","Q27_p","Q5_p","Q3_p")
>
> cols_S <- c("Q1_p","Q4_p","Q7_p","Q9_p","Q21_p",
+            "Q24_p","Q28_p","Q16_p","Q13_p","Q20_p")

```

```

>
> cols_E <- c("Q2_p", "Q4_p", "Q12_p", "Q14_p", "Q15_p",
+             "Q18_p", "Q19_p", "Q26_p", "Q27_p", "Q28_p")
>
> cols_C <- c("Q6_p", "Q7_p", "Q8_p", "Q11_p", "Q12_p",
+             "Q18_p", "Q20_p", "Q23_p", "Q25_p", "Q29_p")
>
> score_dim <- data.frame(
+   ID = ID,
+   R = rowMeans(resp[cols_R], na.rm = TRUE),
+   I = rowMeans(resp[cols_I], na.rm = TRUE),
+   A = rowMeans(resp[cols_A], na.rm = TRUE),
+   S = rowMeans(resp[cols_S], na.rm = TRUE),
+   E = rowMeans(resp[cols_E], na.rm = TRUE),
+   C = rowMeans(resp[cols_C], na.rm = TRUE)
+ )
>
> omega_R <- omega(resp[cols_R], nfactors = 1)

```

Omega_h for 1 factor is not meaningful, just omega_t

Warning message:

In schmid(m, nfactors, fm, digits, rotate = rotate, n.obs = n.obs, :

Omega_h and Omega_asymptotic are not meaningful with one factor> omega_I <-
omega(resp[cols_I], nfactors = 1)

Omega_h for 1 factor is not meaningful, just omega_t

Warning message:

In schmid(m, nfactors, fm, digits, rotate = rotate, n.obs = n.obs, :

Omega_h and Omega_asymptotic are not meaningful with one factor> omega_A <-
omega(resp[cols_A], nfactors = 1)

Omega_h for 1 factor is not meaningful, just omega_t

Warning message:

In schmid(m, nfactors, fm, digits, rotate = rotate, n.obs = n.obs, :

Omega_h and Omega_asymptotic are not meaningful with one factor> omega_S <-
omega(resp[cols_S], nfactors = 1)

Omega_h for 1 factor is not meaningful, just omega_t

Warning message:

In schmid(m, nfactors, fm, digits, rotate = rotate, n.obs = n.obs, :

Omega_h and Omega_asymptotic are not meaningful with one factor> omega_E <-
omega(resp[cols_E], nfactors = 1)

Omega_h for 1 factor is not meaningful, just omega_t

Warning message:

In schmid(m, nfactors, fm, digits, rotate = rotate, n.obs = n.obs, :

Omega_h and Omega_asymptotic are not meaningful with one factor> omega_C <-
omega(resp[cols_C], nfactors = 1)

Omega_h for 1 factor is not meaningful, just omega_t

Warning message:

In schmid(m, nfactors, fm, digits, rotate = rotate, n.obs = n.obs, :

Omega_h and Omega_asymptotic are not meaningful with one factor>

```
> pair_long <- pair_key %>%
+   pivot_longer(cols = c(left, right), values_to = "domain")
>
> table(pair_long$domain)
```

```
  A  C  E  I  R  S
10 10 10 10 10 10
```

```
>
> omega_values <- data.frame(
+   Domain = c("R","I","A","S","E","C"),
+   Omega_T = c(
+     omega_R$omega.tot,
+     omega_I$omega.tot,
+     omega_A$omega.tot,
+     omega_S$omega.tot,
+     omega_E$omega.tot,
+     omega_C$omega.tot
+   )
+ )
```

```
>
> omega_values
  Domain  Omega_T
1      R 0.5026028
2      I 0.5303379
3      A 0.4553248
4      S 0.6268178
5      E 0.7010993
6      C 0.7538567
```

```
>
> omega_values$N_item <- c(
+   length(cols_R),
+   length(cols_I),
+   length(cols_A),
+   length(cols_S),
+   length(cols_E),
+   length(cols_C)
+ )
```

```
>
> omega_values
  Domain  Omega_T N_item
1      R 0.5026028     10
2      I 0.5303379     10
3      A 0.4553248     10
4      S 0.6268178     10
5      E 0.7010993     10
6      C 0.7538567     10
```

```
>
> library(dplyr)
> library(tidyr)
>
> pair_long <- data.frame(
```

```

+   ID = rep(ID, times = nrow(pair_key)),
+   item = rep(pair_key$item, each = length(ID)),
+   left = rep(pair_key$left, each = length(ID)),
+   right = rep(pair_key$right, each = length(ID)),
+   score = unlist(resp[paste0(pair_key$item, "_p")])
+ )
>
> pair_long <- pair_long %>%
+   mutate(
+     winner = ifelse(score == 1, left,
+                     ifelse(score == 0, right, NA))
+   )
>
> domains <- c("R","I","A","S","E","C")
>
> win_matrix <- matrix(0,
+                       nrow = 6,
+                       ncol = 6,
+                       dimnames = list(domains, domains))
>
> for (i in 1:nrow(pair_long)) {
+   w <- pair_long$winner[i]
+   l <- ifelse(pair_long$left[i] == w,
+               pair_long$right[i],
+               pair_long$left[i])
+
+   if (!is.na(w)) {
+     win_matrix[w, l] <- win_matrix[w, l] + 1
+   }
+ }
>
> win_matrix
  R   I   A   S   E   C
R  0 329 838 578 673 189
I 713  0 801 387 848 473
A 204 241  0 721 502 399
S 464 655 321  0 602 476
E 369 194 540 440  0 448
C 853 569 643 566 594  0
>
> prop_matrix <- matrix(NA,
+                       nrow = 6,
+                       ncol = 6,
+                       dimnames = list(domains, domains))
>
> for (i in domains) {
+   for (j in domains) {
+     if (i != j) {
+       wij <- win_matrix[i, j]
+       wji <- win_matrix[j, i]
+       prop_matrix[i, j] <- wij / (wij + wji)
+     }
+   }
+ }

```

```

>
> round(prop_matrix, 3)
      R      I      A      S      E      C
R    NA 0.316 0.804 0.555 0.646 0.181
I 0.684    NA 0.769 0.371 0.814 0.454
A 0.196 0.231    NA 0.692 0.482 0.383
S 0.445 0.629 0.308    NA 0.578 0.457
E 0.354 0.186 0.518 0.422    NA 0.430
C 0.819 0.546 0.617 0.543 0.570    NA
>
> z_matrix <- qnorm(prop_matrix)
>
> thurstone_scale <- rowMeans(z_matrix, na.rm = TRUE)
>
> thurstone_scale
      R      I      A      S
-0.004240648 0.332490993 -0.286723274 -0.044624155
      E      C
-0.318632817 0.321729901
>

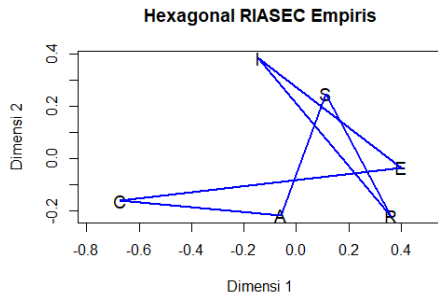
> D <- as.matrix(abs(outer(thurstone_scale,
+                          thurstone_scale,
+                          "-")))
>
> D_scaled <- D / max(D)
>
> R_emp <- 1 - D_scaled
>
> diag(R_emp) <- 1
>
> round(R_emp, 3)
      R      I      A      S      E      C
R 1.000 0.483 0.566 0.938 0.517 0.499
I 0.483 1.000 0.049 0.421 0.000 0.983
A 0.566 0.049 1.000 0.628 0.951 0.066
S 0.938 0.421 0.628 1.000 0.579 0.437
E 0.517 0.000 0.951 0.579 1.000 0.017
C 0.499 0.983 0.066 0.437 0.017 1.000
>
> dist_emp <- as.dist(D)
>
> mds <- cmdscale(dist_emp, k=2, eig=TRUE)
>
> coords <- as.data.frame(mds$points)
> colnames(coords) <- c("Dim1", "Dim2")
> coords$Domain <- names(thurstone_scale)
>
> coords
      Dim1      Dim2 Domain
R 0.004240648 2.521019e-09      R
I -0.332490993 6.921290e-10      I
A 0.286723274 3.111059e-10      A
S 0.044624155 5.072936e-10      S

```

```

E 0.318632817 9.347179e-10      E
C -0.321729901 5.912871e-10    C
>
> dist_emp <- as.dist(abs(outer(thurstone_scale,
+                               thurstone_scale,
+                               "-")))
>
> R_emp <- matrix(c(
+   1.00, 0.483, 0.566, 0.983, 0.517, 0.499,
+   0.483, 1.00, 0.049, 0.421, 0.00, 0.983,
+   0.566, 0.049, 1.00, 0.628, 0.951, 0.066,
+   0.983, 0.421, 0.628, 1.00, 0.579, 0.437,
+   0.517, 0.00, 0.951, 0.579, 1.00, 0.017,
+   0.499, 0.983, 0.066, 0.437, 0.017, 1.00
+ ), nrow = 6, byrow = TRUE)
>
> colnames(R_emp) <- rownames(R_emp) <- c("R","I","A","S","E","C")
>
> R_ideal <- matrix(c(
+   1, 1, .5, 0, .5, 1,
+   1, 1, 1, .5, 0, .5,
+   .5, 1, 1, 1, .5, 0,
+   0, .5, 1, 1, 1, .5,
+   .5, 0, .5, 1, 1, 1,
+   1, .5, 0, .5, 1, 1
+ ), nrow = 6, byrow = TRUE)
>
> colnames(R_ideal) <- rownames(R_ideal) <- c("R","I","A","S","E","C")
>
> upper_emp <- R_emp[upper.tri(R_emp)]
> upper_ideal <- R_ideal[upper.tri(R_ideal)]
>
> CI <- cor(upper_emp, upper_ideal)
> CI
[1] -0.06316006
>
> theta <- seq(0, 2*pi, length.out = 7)[-7]
>
> ideal_coords <- data.frame(
+   x = cos(theta),
+   y = sin(theta),
+   Domain = c("R","I","A","S","E","C")
+ )
>
> emp_coords <- data.frame(
+   x = c(0.35, 0.25, -0.10, 0.15, 0.28, -0.55),
+   y = c(0.25, 0.05, 0.22, -0.18, 0.02, 0.12),
+   Domain = c("R","I","A","S","E","C")
+ )
>
> par(mfrow = c(1,2))
>

```



4. DATA JABATAN MANAJERIAL

R version 4.5.2 (2025-10-31 ucrt) -- "[Not] Part in a Rumble"
 Copyright (C) 2025 The R Foundation for Statistical Computing
 Platform: x86_64-w64-mingw32/x64

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.

```
> install.packages(c("readxl","dplyr","tidyr","psych","GPARotation"))
```

WARNING: Rtools is required to build R packages but is not currently installed. Please download and install the appropriate version of Rtools before proceeding:

<https://cran.rstudio.com/bin/windows/Rtools/>

Installing packages into 'C:/Users/MyPC One Pro L/AppData/Local/R/win-library/4.5'

```
(as 'lib' is unspecified)trying URL
'https://cran.rstudio.com/bin/windows/contrib/4.5/readxl_1.4.5.zip'
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.5/dplyr_1.2.0.zip'
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.5/tidyr_1.3.2.zip'
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.5/psych_2.6.1.zip'
trying URL
'https://cran.rstudio.com/bin/windows/contrib/4.5/GPARotation_2025.3-1.zip'
package 'readxl' successfully unpacked and MD5 sums checked
package 'dplyr' successfully unpacked and MD5 sums checked
package 'tidyr' successfully unpacked and MD5 sums checked
package 'psych' successfully unpacked and MD5 sums checked
```

package 'GPArotation' successfully unpacked and MD5 sums checked

The downloaded binary packages are in

C:\Users\MyPC One Pro

L\AppData\Local\Temp\RtmpaqIV2s\downloaded_packages

```
> library(readxl)
```

```
> library(dplyr)
```

Attaching package: 'dplyr'

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

filter, lag

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

```
intersect, setdiff, setequal, union> library(tidyr)
```

```
> library(psych)
```

```
> library(GPArotation)
```

Attaching package: 'GPArotation'

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

```
equamax, varimin>
> library(readxl)
> manajerial <- read_excel("manajerial.xlsx")
> View(manajerial)
> raw <- read_excel("manajerial.xlsx")
>
> ID <- raw[[1]]
> resp <- raw[, -1]
>
> pair_key <- data.frame(
+   item = paste0("Q", 1:30),
+   left = c("I","A","R","S","R","I","S","A","R","I",
+           "R","E","A","I","R","A","R","E","I","S",
+           "R","I","R","I","A","R","A","S","I","R"),
+   right = c("S","E","I","E","A","C","C","C","S","A",
+           "C","C","S","E","E","S","A","C","E","C",
+           "S","A","C","S","C","E","E","E","C","I"),
+   stringsAsFactors = FALSE
+ )
>
> score_pair <- function(answer, left, right) {
+   ifelse(answer == left, 1,
+         ifelse(answer == right, 0, NA))
+ }
```

```

>
> for (i in 1:nrow(pair_key)) {
+   q <- pair_key$item[i]
+   lp <- pair_key$left[i]
+   rp <- pair_key$right[i]
+
+   resp[[paste0(q, "_p")]] <- score_pair(resp[[q]], lp, rp)
+ }
>
> cols_R <- c("Q3_p", "Q5_p", "Q9_p", "Q11_p", "Q15_p",
+             "Q17_p", "Q21_p", "Q23_p", "Q26_p", "Q30_p")
>
> cols_I <- c("Q1_p", "Q6_p", "Q10_p", "Q14_p", "Q19_p",
+             "Q22_p", "Q24_p", "Q29_p", "Q2_p", "Q12_p")
>
> cols_A <- c("Q2_p", "Q8_p", "Q13_p", "Q16_p", "Q17_p",
+             "Q22_p", "Q25_p", "Q27_p", "Q5_p", "Q3_p")
>
> cols_S <- c("Q1_p", "Q4_p", "Q7_p", "Q9_p", "Q21_p",
+             "Q24_p", "Q28_p", "Q16_p", "Q13_p", "Q20_p")
>
> cols_E <- c("Q2_p", "Q4_p", "Q12_p", "Q14_p", "Q15_p",
+             "Q18_p", "Q19_p", "Q26_p", "Q27_p", "Q28_p")
>
> cols_C <- c("Q6_p", "Q7_p", "Q8_p", "Q11_p", "Q12_p",
+             "Q18_p", "Q20_p", "Q23_p", "Q25_p", "Q29_p")
>
> score_dim <- data.frame(
+   ID = ID,
+   R = rowMeans(resp[cols_R], na.rm = TRUE),
+   I = rowMeans(resp[cols_I], na.rm = TRUE),
+   A = rowMeans(resp[cols_A], na.rm = TRUE),
+   S = rowMeans(resp[cols_S], na.rm = TRUE),
+   E = rowMeans(resp[cols_E], na.rm = TRUE),
+   C = rowMeans(resp[cols_C], na.rm = TRUE)
+ )
>
> omega_R <- omega(resp[cols_R], nfactors = 1)
Omega_h for 1 factor is not meaningful, just omega_t
Warning message:
In schmid(m, nfactors, fm, digits, rotate = rotate, n.obs = n.obs, :
  Omega_h and Omega_asymptotic are not meaningful with one factor> omega_I <-
omega(resp[cols_I], nfactors = 1)
Omega_h for 1 factor is not meaningful, just omega_t
Warning message:
In schmid(m, nfactors, fm, digits, rotate = rotate, n.obs = n.obs, :
  Omega_h and Omega_asymptotic are not meaningful with one factor> omega_A <-
omega(resp[cols_A], nfactors = 1)
Omega_h for 1 factor is not meaningful, just omega_t
Warning messages:

```

```

1: In fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate = rotate, :
  An ultra-Heywood case was detected. Examine the results carefully
2: In schmid(m, nfactors, fm, digits, rotate = rotate, n.obs = n.obs, :
  Omega_h and Omega_asymptotic are not meaningful with one factor> omega_S <-
  omega(resp[cols_S], nfactors = 1)
Omega_h for 1 factor is not meaningful, just omega_t
warning message:
In schmid(m, nfactors, fm, digits, rotate = rotate, n.obs = n.obs, :
  Omega_h and Omega_asymptotic are not meaningful with one factor> omega_E <-
  omega(resp[cols_E], nfactors = 1)
Omega_h for 1 factor is not meaningful, just omega_t
warning message:
In schmid(m, nfactors, fm, digits, rotate = rotate, n.obs = n.obs, :
  Omega_h and Omega_asymptotic are not meaningful with one factor> omega_C <-
  omega(resp[cols_C], nfactors = 1)
Omega_h for 1 factor is not meaningful, just omega_t
warning message:
In schmid(m, nfactors, fm, digits, rotate = rotate, n.obs = n.obs, :
  Omega_h and Omega_asymptotic are not meaningful with one factor>
> omega_R <- omega(resp[cols_R], nfactors = 1)
Omega_h for 1 factor is not meaningful, just omega_t
warning message:
In schmid(m, nfactors, fm, digits, rotate = rotate, n.obs = n.obs, :
  Omega_h and Omega_asymptotic are not meaningful with one factor> omega_I <-
  omega(resp[cols_I], nfactors = 1)
Omega_h for 1 factor is not meaningful, just omega_t
warning message:
In schmid(m, nfactors, fm, digits, rotate = rotate, n.obs = n.obs, :
  Omega_h and Omega_asymptotic are not meaningful with one factor> omega_A <-
  omega(resp[cols_A], nfactors = 1)
Omega_h for 1 factor is not meaningful, just omega_t
warning messages:
1: In fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate = rotate, :
  An ultra-Heywood case was detected. Examine the results carefully
2: In schmid(m, nfactors, fm, digits, rotate = rotate, n.obs = n.obs, :
  Omega_h and Omega_asymptotic are not meaningful with one factor> omega_S <-
  omega(resp[cols_S], nfactors = 1)
Omega_h for 1 factor is not meaningful, just omega_t
warning message:
In schmid(m, nfactors, fm, digits, rotate = rotate, n.obs = n.obs, :

```

Omega_h and Omega_asymptotic are not meaningful with one factor> `omega_E <- omega(resp[cols_E], nfactors = 1)`

Omega_h for 1 factor is not meaningful, just omega_t

Warning message:

In `schmid(m, nfactors, fm, digits, rotate = rotate, n.obs = n.obs, :`

Omega_h and Omega_asymptotic are not meaningful with one factor> `omega_C <- omega(resp[cols_C], nfactors = 1)`

Omega_h for 1 factor is not meaningful, just omega_t

Warning message:

In `schmid(m, nfactors, fm, digits, rotate = rotate, n.obs = n.obs, :`

Omega_h and Omega_asymptotic are not meaningful with one factor>

> `pair_long <- pair_key %>%`

+ `pivot_longer(cols = c(left, right), values_to = "domain")`

>

> `table(pair_long$domain)`

```
A C E I R S
10 10 10 10 10 10
```

>

> `omega_values <- data.frame(`

+ `Domain = c("R","I","A","S","E","C"),`

+ `Omega_T = c(`

+ `omega_R$omega.tot,`

+ `omega_I$omega.tot,`

+ `omega_A$omega.tot,`

+ `omega_S$omega.tot,`

+ `omega_E$omega.tot,`

+ `omega_C$omega.tot`

+ `)`

+ `)`

>

> `omega_values`

```
Domain Omega_T
```

```
1      R 0.4762556
```

```
2      I 0.5184294
```

```
3      A 0.2937325
```

```
4      S 0.6324324
```

```
5      E 0.5918793
```

```
6      C 0.7221642
```

>

> `omega_values$N_item <- c(`

+ `length(cols_R),`

+ `length(cols_I),`

+ `length(cols_A),`

+ `length(cols_S),`

+ `length(cols_E),`

+ `length(cols_C)`

+ `)`

>

> `omega_values`

```
Domain Omega_T N_item
```

```

1      R 0.4762556      10
2      I 0.5184294      10
3      A 0.2937325      10
4      S 0.6324324      10
5      E 0.5918793      10
6      C 0.7221642      10
>
> library(dplyr)
> library(tidyr)
>
> pair_long <- data.frame(
+   ID = rep(ID, times = nrow(pair_key)),
+   item = rep(pair_key$item, each = length(ID)),
+   left = rep(pair_key$left, each = length(ID)),
+   right = rep(pair_key$right, each = length(ID)),
+   score = unlist(resp[paste0(pair_key$item, "_p")])
+ )
>
> pair_long <- pair_long %>%
+   mutate(
+     winner = ifelse(score == 1, left,
+                     ifelse(score == 0, right, NA))
+   )
>
> domains <- c("R","I","A","S","E","C")
>
> win_matrix <- matrix(0,
+                       nrow = 6,
+                       ncol = 6,
+                       dimnames = list(domains, domains))
>
> for (i in 1:nrow(pair_long)) {
+   w <- pair_long$winner[i]
+   l <- ifelse(pair_long$left[i] == w,
+               pair_long$right[i],
+               pair_long$left[i])
+
+   if (!is.na(w)) {
+     win_matrix[w, l] <- win_matrix[w, l] + 1
+   }
+ }
>
> win_matrix
      R      I      A      S      E      C
R      0    515 1381   825 1039 301
I 1155      0 1377   529 1343 660
A  289  293      0 1099   664 725
S  845 1141  571      0   862 848
E  631  327 1006   808      0 885
C 1369 1010  945   822   785   0
>
> prop_matrix <- matrix(NA,
+                       nrow = 6,
+                       ncol = 6,

```

```

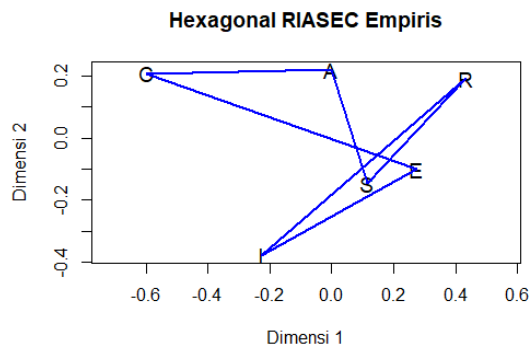
+                 dimnames = list(domains, domains))
>
> for (i in domains) {
+   for (j in domains) {
+     if (i != j) {
+       wij <- win_matrix[i, j]
+       wji <- win_matrix[j, i]
+       prop_matrix[i, j] <- wij / (wij + wji)
+     }
+   }
+ }
>
> round(prop_matrix, 3)
      R      I      A      S      E      C
R    NA 0.308 0.827 0.494 0.622 0.180
I 0.692    NA 0.825 0.317 0.804 0.395
A 0.173 0.175    NA 0.658 0.398 0.434
S 0.506 0.683 0.342    NA 0.516 0.508
E 0.378 0.196 0.602 0.484    NA 0.530
C 0.820 0.605 0.566 0.492 0.470    NA
>
> z_matrix <- qnorm(prop_matrix)
>
> thurstone_scale <- rowMeans(z_matrix, na.rm = TRUE)
>
> thurstone_scale
      R      I      A      S
-0.03531765 0.30948987 -0.37863829 0.02891652
      E      C
-0.17474054 0.25029008
>
> D <- as.matrix(abs(outer(thurstone_scale,
+                          thurstone_scale,
+                          "-")))
>
> D_scaled <- D / max(D)
>
> R_emp <- 1 - D_scaled
>
> diag(R_emp) <- 1
>
> round(R_emp, 3)
      R      I      A      S      E      C
R 1.000 0.499 0.501 0.907 0.797 0.585
I 0.499 1.000 0.000 0.592 0.296 0.914
A 0.501 0.000 1.000 0.408 0.704 0.086
S 0.907 0.592 0.408 1.000 0.704 0.678
E 0.797 0.296 0.704 0.704 1.000 0.382
C 0.585 0.914 0.086 0.678 0.382 1.000
>
> dist_emp <- as.dist(D)
>
> mds <- cmdscale(dist_emp, k=2, eig=TRUE)
>

```

```

> coords <- as.data.frame(mds$points)
> colnames(coords) <- c("Dim1","Dim2")
> coords$Domain <- names(thurstone_scale)
>
> coords
      Dim1      Dim2 Domain
R  0.03531765  2.423980e-09    R
I -0.30948987 -2.760502e-09    I
A  0.37863829 -1.342125e-09    A
S -0.02891652 -3.923855e-10    S
E  0.17474054  5.147816e-10    E
C -0.25029008  2.129834e-09    C
>
> thurstone_scale
      R      I      A      S
-0.03531765  0.30948987 -0.37863829  0.02891652
      E      C
-0.17474054  0.25029008
>
> D <- as.matrix(abs(outer(thurstone_scale,
+                          thurstone_scale,
+                          "-")))
>
> round(D,3)
      R      I      A      S      E      C
R 0.000 0.345 0.343 0.064 0.139 0.286
I 0.345 0.000 0.688 0.281 0.484 0.059
A 0.343 0.688 0.000 0.408 0.204 0.629
S 0.064 0.281 0.408 0.000 0.204 0.221
E 0.139 0.484 0.204 0.204 0.000 0.425
C 0.286 0.059 0.629 0.221 0.425 0.000
>
> dist_emp <- as.dist(D)
>
> mds <- cmdscale(dist_emp, k = 2, eig = TRUE)
>
> score_dim <- as.data.frame(mds$points)
>
> colnames(score_dim) <- c("Dim1","Dim2")
>
> score_dim$Domain <- names(thurstone_scale)
>
> score_dim
      Dim1      Dim2 Domain
R  0.03531765  2.423980e-09    R
I -0.30948987 -2.760502e-09    I
A  0.37863829 -1.342125e-09    A
S -0.02891652 -3.923855e-10    S
E  0.17474054  5.147816e-10    E
C -0.25029008  2.129834e-09    C
>

```



>

5. DATA JABATAN PELAKSANA

R version 4.5.2 (2025-10-31 ucrt) -- "[Not] Part in a Rumble"
 Copyright (C) 2025 The R Foundation for Statistical Computing
 Platform: x86_64-w64-mingw32/x64

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.

```
> install.packages(c("readxl","dplyr","tidyr","psych","GPARotation"))
```

WARNING: Rtools is required to build R packages but is not currently installed. Please download and install the appropriate version of Rtools before proceeding:

<https://cran.rstudio.com/bin/windows/Rtools/>

Installing packages into 'C:/Users/MyPC One Pro L/AppData/Local/R/win-library/4.5'

```
(as 'lib' is unspecified)trying URL
'https://cran.rstudio.com/bin/windows/contrib/4.5/readxl_1.4.5.zip'
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.5/dplyr_1.2.0.zip'
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.5/tidyr_1.3.2.zip'
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.5/psych_2.6.1.zip'
trying URL
'https://cran.rstudio.com/bin/windows/contrib/4.5/GPARotation_2025.3-1.zip'
```

```
package 'readxl' successfully unpacked and MD5 sums checked
package 'dplyr' successfully unpacked and MD5 sums checked
package 'tidyr' successfully unpacked and MD5 sums checked
package 'psych' successfully unpacked and MD5 sums checked
package 'GPARotation' successfully unpacked and MD5 sums checked
```

```
The downloaded binary packages are in
  C:\Users\MyPC One Pro
L\AppData\Local\Temp\RtmpcfACM9\downloaded_packages
> library(readxl)
> library(dplyr)
```

Attaching package: 'dplyr'

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

```
filter, lag
```

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

```
intersect, setdiff, setequal, union> library(tidyr)
> library(psych)
> library(GPARotation)
```

Attaching package: 'GPARotation'

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

```
equamax, varimin>
> library(readxl)
> pelaksana <- read_excel("pelaksana.xlsx")
> View(pelaksana)
> raw <- read_excel("pelaksana.xlsx")
>
> ID <- raw[[1]]
> resp <- raw[, -1]
>
> pair_key <- data.frame(
+   item = paste0("Q", 1:30),
+   left = c("I","A","R","S","R","I","S","A","R","I",
+            "R","E","A","I","R","A","R","E","I","S",
+            "R","I","R","I","A","R","A","S","I","R"),
+   right = c("S","E","I","E","A","C","C","C","S","A",
+            "C","C","S","E","E","S","A","C","E","C",
+            "S","A","C","S","C","E","E","E","C","I"),
+   stringsAsFactors = FALSE
+ )
>
```

```
> score_pair <- function(answer, left, right) {
+   ifelse(answer == left, 1,
+         ifelse(answer == right, 0, NA))
+ }
```

```
> Generalisir ke semua nomor
```

```
Error: unexpected symbol in "Generalisir ke"
```

```
> score_pair <- function(answer, left, right) {
+   ifelse(answer == left, 1,
+         ifelse(answer == right, 0, NA))
+ }
>
> for (i in 1:nrow(pair_key)) {
+   q <- pair_key$item[i]
+   lp <- pair_key$left[i]
+   rp <- pair_key$right[i]
+   resp[[paste0(q, "_p")]] <- score_pair(resp[[q]], lp, rp)
+ }
>
> cols_R <- c("Q3_p", "Q5_p", "Q9_p", "Q11_p", "Q15_p",
+             "Q17_p", "Q21_p", "Q23_p", "Q26_p", "Q30_p")
>
> cols_I <- c("Q1_p", "Q6_p", "Q10_p", "Q14_p", "Q19_p",
+             "Q22_p", "Q24_p", "Q29_p", "Q2_p", "Q12_p")
>
> cols_A <- c("Q2_p", "Q8_p", "Q13_p", "Q16_p", "Q17_p",
+             "Q22_p", "Q25_p", "Q27_p", "Q5_p", "Q3_p")
>
> cols_S <- c("Q1_p", "Q4_p", "Q7_p", "Q9_p", "Q21_p",
+             "Q24_p", "Q28_p", "Q16_p", "Q13_p", "Q20_p")
>
> cols_E <- c("Q2_p", "Q4_p", "Q12_p", "Q14_p", "Q15_p",
+             "Q18_p", "Q19_p", "Q26_p", "Q27_p", "Q28_p")
>
> cols_C <- c("Q6_p", "Q7_p", "Q8_p", "Q11_p", "Q12_p",
+             "Q18_p", "Q20_p", "Q23_p", "Q25_p", "Q29_p")
>
>
> score_dim <- data.frame(
+   ID = ID,
+   R = rowMeans(resp[cols_R], na.rm = TRUE),
+   I = rowMeans(resp[cols_I], na.rm = TRUE),
+   A = rowMeans(resp[cols_A], na.rm = TRUE),
+   S = rowMeans(resp[cols_S], na.rm = TRUE),
+   E = rowMeans(resp[cols_E], na.rm = TRUE),
+   C = rowMeans(resp[cols_C], na.rm = TRUE)
+ )
>
> omega_R <- omega(resp[cols_R], nfactors = 1)
```

Omega_h for 1 factor is not meaningful, just omega_t

warning message:

```

In schmid(m, nfactors, fm, digits, rotate = rotate, n.obs = n.obs, :
  Omega_h and Omega_asymptotic are not meaningful with one factor> omega_I <-
  omega(resp[cols_I], nfactors = 1)
Omega_h for 1 factor is not meaningful, just omega_t
Warning message:
In schmid(m, nfactors, fm, digits, rotate = rotate, n.obs = n.obs, :
  Omega_h and Omega_asymptotic are not meaningful with one factor> omega_A <-
  omega(resp[cols_A], nfactors = 1)
Omega_h for 1 factor is not meaningful, just omega_t
Warning message:
In schmid(m, nfactors, fm, digits, rotate = rotate, n.obs = n.obs, :
  Omega_h and Omega_asymptotic are not meaningful with one factor> omega_S <-
  omega(resp[cols_S], nfactors = 1)
Omega_h for 1 factor is not meaningful, just omega_t
Warning message:
In schmid(m, nfactors, fm, digits, rotate = rotate, n.obs = n.obs, :
  Omega_h and Omega_asymptotic are not meaningful with one factor> omega_E <-
  omega(resp[cols_E], nfactors = 1)
Omega_h for 1 factor is not meaningful, just omega_t
Warning message:
In schmid(m, nfactors, fm, digits, rotate = rotate, n.obs = n.obs, :
  Omega_h and Omega_asymptotic are not meaningful with one factor> omega_C <-
  omega(resp[cols_C], nfactors = 1)
Omega_h for 1 factor is not meaningful, just omega_t
Warning message:
In schmid(m, nfactors, fm, digits, rotate = rotate, n.obs = n.obs, :
  Omega_h and Omega_asymptotic are not meaningful with one factor>
>
> pair_long <- pair_key %>%
+   pivot_longer(cols = c(left, right), values_to = "domain")
>
> table(pair_long$domain)

  A  C  E  I  R  S
10 10 10 10 10 10
>
> omega_values <- data.frame(
+   Domain = c("R", "I", "A", "S", "E", "C"),
+   Omega_T = c(
+     omega_R$omega.tot,
+     omega_I$omega.tot,
+     omega_A$omega.tot,
+     omega_S$omega.tot,
+     omega_E$omega.tot,
+     omega_C$omega.tot
+   )

```

```

+ )
>
> omega_values
  Domain   Omega_T
1      R 0.4135233
2      I 0.5371466
3      A 0.4395341
4      S 0.6361773
5      E 0.6135958
6      C 0.7385178
>
> omega_values$N_item <- c(
+   length(cols_R),
+   length(cols_I),
+   length(cols_A),
+   length(cols_S),
+   length(cols_E),
+   length(cols_C)
+ )
>
> omega_values
  Domain   Omega_T N_item
1      R 0.4135233    10
2      I 0.5371466    10
3      A 0.4395341    10
4      S 0.6361773    10
5      E 0.6135958    10
6      C 0.7385178    10
>
> library(dplyr)
> library(tidyr)
>
> pair_long <- data.frame(
+   ID = rep(ID, times = nrow(pair_key)),
+   item = rep(pair_key$item, each = length(ID)),
+   left = rep(pair_key$left, each = length(ID)),
+   right = rep(pair_key$right, each = length(ID)),
+   score = unlist(resp[paste0(pair_key$item, "_p")])
+ )
>
> pair_long <- pair_long %>%
+   mutate(
+     winner = ifelse(score == 1, left,
+                     ifelse(score == 0, right, NA))
+   )
>
> domains <- c("R","I","A","S","E","C")
>
> win_matrix <- matrix(0,
+                       nrow = 6,
+                       ncol = 6,
+                       dimnames = list(domains, domains))
>
> for (i in 1:nrow(pair_long)) {

```

```

+     w <- pair_long$winner[i]
+     l <- ifelse(pair_long$left[i] == w,
+                 pair_long$right[i],
+                 pair_long$left[i])
+
+     if (!is.na(w)) {
+         win_matrix[w, l] <- win_matrix[w, l] + 1
+     }
+ }
>
> win_matrix
  R   I   A   S   E   C
R  0 477 862 714  782 221
I 711   0 920 493 1003 463
A 326 268   0 761  646 435
S 474 695 427   0  702 445
E 406 185 542 486   0 395
C 967 725 753 743  793   0
>
> prop_matrix <- matrix(NA,
+                        nrow = 6,
+                        ncol = 6,
+                        dimnames = list(domains, domains))
>
> for (i in domains) {
+   for (j in domains) {
+     if (i != j) {
+       wij <- win_matrix[i, j]
+       wji <- win_matrix[j, i]
+       prop_matrix[i, j] <- wij / (wij + wji)
+     }
+   }
+ }
>
> round(prop_matrix, 3)
  R   I   A   S   E   C
R  NA 0.402 0.726 0.601 0.658 0.186
I 0.598   NA 0.774 0.415 0.844 0.390
A 0.274 0.226   NA 0.641 0.544 0.366
S 0.399 0.585 0.359   NA 0.591 0.375
E 0.342 0.156 0.456 0.409   NA 0.332
C 0.814 0.610 0.634 0.625 0.668   NA
>
> z_matrix <- qnorm(prop_matrix)
>
>
> thurstone_scale <- rowMeans(z_matrix, na.rm = TRUE)
>
> thurstone_scale
  R           I           A           S
0.02422378 0.30406039 -0.24501751 -0.09821446
  E           C
-0.43854893 0.45349673
>

```

```

> D <- as.matrix(abs(outer(thurstone_scale,
+                           thurstone_scale,
+                           "-")))
>
> D_scaled <- D / max(D)
>
> R_emp <- 1 - D_scaled
>
> diag(R_emp) <- 1
>
> round(R_emp, 3)
      R      I      A      S      E      C
R 1.000 0.686 0.698 0.863 0.481 0.519
I 0.686 1.000 0.384 0.549 0.168 0.832
A 0.698 0.384 1.000 0.835 0.783 0.217
S 0.863 0.549 0.835 1.000 0.618 0.382
E 0.481 0.168 0.783 0.618 1.000 0.000
C 0.519 0.832 0.217 0.382 0.000 1.000
>
> R_emp <- matrix(c(
+   1.00, 0.686, 0.698, 0.863, 0.481, 0.519,
+   0.686, 1.00, 0.384, 0.549, 0.168, 0.832,
+   0.698, 0.384, 1.00, 0.835, 0.783, 0.217,
+   0.863, 0.549, 0.835, 1.00, 0.618, 0.382,
+   0.481, 0.168, 0.783, 0.618, 1.00, 0.000,
+   0.519, 0.832, 0.217, 0.382, 0.000, 1.00
+ ), nrow = 6, byrow = TRUE)
>
> colnames(R_emp) <- rownames(R_emp) <- c("R","I","A","S","E","C")
>
> R_emp
      R      I      A      S      E      C
R 1.000 0.686 0.698 0.863 0.481 0.519
I 0.686 1.000 0.384 0.549 0.168 0.832
A 0.698 0.384 1.000 0.835 0.783 0.217
S 0.863 0.549 0.835 1.000 0.618 0.382
E 0.481 0.168 0.783 0.618 1.000 0.000
C 0.519 0.832 0.217 0.382 0.000 1.000
>

> R_ideal <- matrix(c(
+   1, 1, .5, 0, .5, 1,
+   1, 1, 1, .5, 0, .5,
+   .5, 1, 1, 1, .5, 0,
+   0, .5, 1, 1, 1, .5,
+   .5, 0, .5, 1, 1, 1,
+   1, .5, 0, .5, 1, 1
+ ), nrow = 6, byrow = TRUE)
>
> colnames(R_ideal) <- rownames(R_ideal) <- c("R","I","A","S","E","C")
>
>
> upper_emp <- R_emp[upper.tri(R_emp)]
> upper_ideal <- R_ideal[upper.tri(R_ideal)]

```

```

>
> CI <- cor(upper_emp, upper_ideal)
> CI
[1] 0.06673514
>
> thurstone_scale
      R          I          A          S
0.02422378 0.30406039 -0.24501751 -0.09821446
      E          C
-0.43854893 0.45349673
> D <- as.matrix(abs(outer(thurstone_scale,
+                          thurstone_scale,
+                          "-")))
>
> dist_emp <- as.dist(D)
> mds <- cmdscale(dist_emp, k = 2, eig = TRUE)
> score_dim <- as.data.frame(mds$points)
>
> colnames(score_dim) <- c("Dim1", "Dim2")
>
> score_dim$Domain <- names(thurstone_scale)
>
> score_dim
      Dim1      Dim2 Domain
R -0.02422378 -3.867135e-09      R
I -0.30406039 -2.076897e-09      I
A  0.24501751 -5.999865e-09      A
S  0.09821446 -1.336511e-09      S
E  0.43854893  3.091222e-09      E
C -0.45349673  1.057327e-09      C
>
> plot(score_dim$Dim1,
+      score_dim$Dim2,
+      type="n",
+      xlab="Dim1",
+      ylab="Dim2",
+      main="MDS Configuration (Empirical RIASEC)")
>
> text(score_dim$Dim1,
+      score_dim$Dim2,
+      labels=score_dim$Domain,
+      cex=1.2)
>

```

Hexagonal RIASEC Empiris

