# Data Processing and Test of Reliability

**John Robert Torres**

**2025-06-30**

# 1 Load required and new packages

```
if (!require("pacman")) install.packages("pacman")
```

```
## Loading required package: pacman
```

```
library(pacman)
pacman::p_load("here", "glue", "crayon","readxl", "writexl", "dplyr", "tidyr", "rstatix")
pacman::p_load("psych", "aaRon")

`%ni%` = Negate(`%in%`)
```

# 2 Set data paths and details

```
main.path = here::here()
data.path = file.path(main.path, "02 Data")
output.path = file.path(main.path, "04 Outputs")

file.name = "063025_Lawas_Pilot_Data.xlsx"
sheet.name = "FINAL"
client.surname = "LAWAS"
output.name = glue::glue(paste0(format(Sys.Date(), "%m%d%y"), "_{client.surname}.xlsx"))
```

# 3 Load dataset

```
df.raw = readxl::read_excel(file.path(data.path, file.name),
                            sheet = sheet.name,
                            skip = 1)

alpha.raw = readxl::read_excel(file.path(data.path, file.name),
                               sheet = "ALPHA")
```

# 4 Process data

```
df.proc = df.raw %>%
  tidyr::pivot_longer(-c("Respondent"),
                      names_to = "Key",
                      values_to = "Rating") %>%
  tidyr::separate("Key",
                  into = c("Type", "Statement", "Characteristic"),
                  sep = "-")
```

# 5 Implement methodology

# 5.1 Cronbach's Alpha Run

```
alpha.results.1 = alpha.raw %>%
  dplyr::select(-Respondent) %>%
  psych::alpha()

prettyalpha(alpha.results.1, dp = 2)
```

```
##
##
## **Cronbach Alpha**:
##
##
##
## |   | RAW_ALPHA | STD.ALPHA | G6(SMC) | AVERAGE_R | S/N  | ASE  | MEAN |  SD  | MEDIAN_R |
## |:--|:---------:|:---------:|:-------:|:---------:|:----:|:----:|:----:|:----:|:--------:|
## |   |    0.7    |   0.71    |  0.66   |   0.38    | 2.43 | 0.16 |  4   | 0.44 |   0.35   |
##
##
## **Alpha Values If Certain Items Were Dropped**:
##
##
##
## |   | RAW_ALPHA | STD.ALPHA | G6(SMC) | AVERAGE_R | S/N  | ALPHA SE | VAR.R | MED.R |
## |:--|:---------:|:---------:|:-------:|:---------:|:----:|:--------:|:-----:|:-----:|
## |S1 |   0.57    |   0.58    |  0.49   |   0.32    | 1.40 |   0.24   | 0.00  | 0.32  |
## |S2 |   0.72    |   0.73    |  0.65   |   0.47    | 2.65 |   0.15   | 0.01  | 0.49  |
## |S3 |   0.60    |   0.61    |  0.53   |   0.34    | 1.58 |   0.22   | 0.02  | 0.29  |
## |S4 |   0.63    |   0.65    |  0.57   |   0.38    | 1.86 |   0.20   | 0.02  | 0.32  |
##
##
## **Item-Level Statistics**:
##
##
##
## |   | N  | RAW.R | STD.R | R.COR | R.DROP | MEAN |  SD  |
## |:--|:--:|:-----:|:-----:|:-----:|:------:|:----:|:----:|
## |S1 | 10 | 0.78  | 0.79  | 0.72  |  0.59  | 3.9  | 0.57 |
## |S2 | 10 | 0.66  | 0.64  | 0.42  |  0.35  | 4.0  | 0.67 |
## |S3 | 10 | 0.73  | 0.77  | 0.66  |  0.54  | 4.4  | 0.52 |
## |S4 | 10 | 0.75  | 0.73  | 0.59  |  0.48  | 3.7  | 0.67 |
```

```
alpha.raw %>%
  dplyr::select(-Respondent) %>%
  ltm::cronbach.alpha(CI = TRUE)
```

```
##
## Cronbach's alpha for the '.' data-set
##
## Items: 4
## Sample units: 10
## alpha: 0.695
##
## Bootstrap 95% CI based on 1000 samples
##   2.5%  97.5%
## -0.619  0.901
```

# 5.2 Cronbach's Alpha Final

```
alpha.total = as.data.frame(alpha.results.1$total)

alpha.item = as.data.frame(alpha.results.1$item.stats) %>%
  tibble::rownames_to_column("item")

alpha.drop = as.data.frame(alpha.results.1$alpha.drop) %>%
  tibble::rownames_to_column("item")
```

# 6 Export necessary data

```
export.list = list(DATA = df.proc,
                   ALPHA.TOTAL = alpha.total,
                   ALPHA.ITEM = alpha.item,
                   ALPHA.DROP = alpha.drop)

if(length(export.list) != 0){
  if (!file.exists(file.path(output.path, output.name))) {
    writexl::write_xlsx(export.list, file.path(output.path, output.name))
    cat(crayon::green("File successfully written."))
  } else {
    cat(crayon::red(glue::glue("Filename already used: {output.name}")))
    overwrite = readline(prompt = "Overwrite (1 for Yes, 0 for No): ")
    if (overwrite == "1") {
      writexl::write_xlsx(export.list, file.path(output.path, output.name))
      cat(crayon::green("File successfully overwritten"))
    } else {
      cat(crayon::red("File not overwritten"))
    }
  }
}
```

```
## Filename already used: 070925_LAWAS.xlsxOverwrite (1 for Yes, 0 for No):
## File not overwritten
```