# Code for replication - 2

Marius Geantă

May 19, 2024

Read the empirical data in original and long format.

```
df_long <- readRDS("df_long.rds") # long format
df <- readRDS("df.rds")
```

Load packages and compute descriptive statistics. These are available as *Table 1*.

```
if (!require(dplyr)) install.packages("knitr")
if (!require(tidyr)) install.packages("tidyverse")
if (!require(dplyr)) install.packages("broom")

library(knitr)
library(tidyverse)
library(broom)
```

```
summary_stats <- df_long %>%
  group_by(tool, criterion) %>%
  summarise(
    Mean = mean(score, na.rm = TRUE),
    Median = median(score, na.rm = TRUE),
    SD = sd(score, na.rm = TRUE),
    .groups = 'drop'
  )
```

```
kable(summary_stats,
      format = "latex",  # Specify LaTeX output for PDF documents
      caption = "Summary Statistics of Scores by Tool and Criterion",
      align = 'c')
```

Next, we conduct the analysis of variance. The corresponding results are available as *Table 2*

```
anova_results <- aov(score ~ tool * criterion, data = df_long)
```

```
anova_table <- broom::tidy(anova_results)
knitr::kable(anova_table,
             caption = "ANOVA Results for Score by Tool and Criterion",
             align = 'c',
             format = "latex")
```

```
if (!require(dplyr)) install.packages("dplyr")
if (!require(tidyr)) install.packages("tidyr")
library(dplyr)
library(tidyr)
```

```
df_wide <- df_long %>%
  pivot_wider(names_from = criterion, values_from = score, values_fn = list(score = mean))
```

Table 1: Summary Statistics of Scores by Tool and Criterion

| tool | criterion | Mean | Median | SD |
|---|---|---|---|---|
| chatgpt | accuracy | 4.130 | 4 | 0.8163530 |
| chatgpt | comprehensiveness | 4.080 | 4 | 0.7723878 |
| chatgpt | friendly | 4.300 | 4 | 0.7957172 |
| chatgpt | timeliness | 4.150 | 4 | 0.8312369 |
| guide | accuracy | 3.510 | 4 | 1.3449702 |
| guide | comprehensiveness | 3.260 | 3 | 1.3717905 |
| guide | friendly | 3.825 | 4 | 1.1922594 |
| guide | timeliness | 3.865 | 4 | 1.0967173 |
| gemini | accuracy | 3.510 | 4 | 1.0074596 |
| gemini | comprehensiveness | 3.225 | 3 | 0.9320507 |
| gemini | friendly | 3.990 | 4 | 0.9242136 |
| gemini | timeliness | 3.745 | 4 | 0.9133158 |
| copilot | accuracy | 3.840 | 4 | 1.0956286 |
| copilot | comprehensiveness | 3.845 | 4 | 0.9827533 |
| copilot | friendly | 4.225 | 4 | 0.8473280 |
| copilot | timeliness | 3.980 | 4 | 0.9294728 |

Table 2: ANOVA Results for Score by Tool and Criterion

| term | df | sumsq | meansq | statistic | p.value |
|---|---|---|---|---|---|
| tool | 3 | 178.63 | 59.543333 | 58.727530 | 0.0000000 |
| criterion | 3 | 107.19 | 35.730000 | 35.240463 | 0.0000000 |
| tool:criterion | 9 | 30.57 | 3.396667 | 3.350129 | 0.0004315 |
| Residuals | 3184 | 3228.23 | 1.013891 | NA | NA |

We continue by performing Manova and computing the Inter Class Correlation (ICC). In this respect, please, see *Table 3* and *Table 4*.

```
if (!require(dplyr)) install.packages("MASS")
if (!require(tidyr)) install.packages("pander")
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 4.2.3
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(pander)
```

```
manova_model <- manova(cbind(accuracy, timeliness, comprehensiveness,
    friendly) ~ tool, data = df_wide)

manova_summary <- summary(manova_model)

pander(manova_summary)
```

| | Df | Pillai | approx F | num Df | den Df | Pr(>F) |
|---|---|---|---|---|---|---|
| **tool** | 3 | 0.1481 | 10.32 | 12 | 2385 | 3.897e-20 |
| **Residuals** | 796 | NA | NA | NA | NA | NA |

```
if (!require(psych)) install.packages("psych")
if (!require(dplyr)) install.packages("dplyr")
if (!require(knitr)) install.packages("knitr")
if (!require(knitr)) install.packages("tidyr")

library(psych)
library(dplyr)
library(knitr)
library(tidyr)
```

```
data_wide <- df_long %>%
  unite("id_criterion", id, criterion, sep = "_") %>%
  pivot_wider(
    names_from = "id_criterion",
    values_from = "score",
    names_prefix = "exp_"
  )
```

```
icc_data <- dplyr::select(data_wide, starts_with("exp"))
icc_results <- ICC(as.matrix(icc_data))
print(icc_results)
```

```
## Call: ICC(x = as.matrix(icc_data))
##
## Intraclass correlation coefficients
```

```
##                          type  ICC  F df1  df2        p lower bound upper bound
## Single_raters_absolute   ICC1 0.24 11  99 3100 4.5e-141        0.19        0.31
## Single_random_raters     ICC2 0.24 14  99 3069 3.9e-182        0.19        0.31
## Single_fixed_raters      ICC3 0.29 14  99 3069 3.9e-182        0.23        0.36
## Average_raters_absolute ICC1k 0.91 11  99 3100 4.5e-141        0.88        0.93
## Average_random_raters   ICC2k 0.91 14  99 3069 3.9e-182        0.88        0.94
## Average_fixed_raters    ICC3k 0.93 14  99 3069 3.9e-182        0.91        0.95
##
##  Number of subjects = 100     Number of Judges =  32
## See the help file for a discussion of the other 4 McGraw and Wong estimates,
```

```
library(lme4)
```

```
# Fit a mixed model: scores are nested within experts and criteria
model <- lmer(score ~ (1|id) + (1|criterion) + (1|tool), data = df_long)
summary(model)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: score ~ (1 | id) + (1 | criterion) + (1 | tool)
##    Data: df_long
##
## REML criterion at convergence: 8801.3
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.9528 -0.5904  0.1199  0.7004  2.4032
##
## Random effects:
##  Groups    Name        Variance Std.Dev.
##  id        (Intercept) 0.13773  0.3711
##  criterion (Intercept) 0.04354  0.2087
##  tool      (Intercept) 0.07330  0.2707
##  Residual              0.89983  0.9486
## Number of obs: 3200, groups:  id, 8; criterion, 4; tool, 4
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   3.8425     0.2161   17.78
```