

# IRAnalysis

Chang Lu

2025-04-04

part 1 kappa by criteria

```
latchman <- read_excel("Latchman_Rubric Scoring Sheet 2025.xlsx", sheet = 1)
kva      <- read_excel("Rubric Scoring Sheet 2025 kva UF.xlsx", sheet = 1)
ly       <- read_excel("Rubric Scoring Sheet 2025 LY UF.xlsx", sheet = 1)
wagner   <- read_excel("Wagner_Rubric Scoring Sheet 2025.xlsx", sheet = 1)
```

```
clean_rater <- function(df) {
  df_clean <- df[1:12, -(1:2)]
  df_clean <- df_clean %>%
    mutate_all(as.character) %>%
    mutate_all(toupper)
  return(df_clean)
}

latchman_clean <- clean_rater(latchman)
kva_clean      <- clean_rater(kva)
ly_clean       <- clean_rater(ly)
wagner_clean   <- clean_rater(wagner)
#tail(latchman)
```

```
normalize_score <- function(x) {
  x <- toupper(trimws(x)) # remove whitespace and make uppercase

  if (x %in% c("N/A", "NA")) {
    return(NA)
  } else if (grepl("^Y(\\b|\\W)", x)) {
    return("Y")
  } else if (grepl("^N(\\b|\\W)", x)) {
    return("N")
  } else {
    return(NA) # comments like "A IS DIFFERENT..." will be treated as NA
  }
}
```

```
latchman_normalized <- latchman_clean %>%
  mutate(across(everything(), ~sapply(., normalize_score)))
kva_normalized <- kva_clean %>%
  mutate(across(everything(), ~sapply(., normalize_score)))
ly_normalized <- ly_clean %>%
```

```

mutate(across(everything(), ~sapply(., normalize_score)))
wagner_normalized <- wagner_clean %>%
mutate(across(everything(), ~sapply(., normalize_score)))
#tail(latchman_normalized)

```

```

reshape_rater <- function(df, rater_name) {
  df <- df %>%
    filter(rowSums(!is.na(.)) > 1) %>% # Drop rows that are all NA or mostly empty
    mutate(Criterion = 1:nrow(.)) %>%
    pivot_longer(
      cols = -Criterion,
      names_to = "Question",
      values_to = "Score"
    ) %>%
    mutate(Rater = rater_name)
  return(df)
}

```

```

latchman_long <- reshape_rater(latchman_normalized, "Latchman")
kva_long <- reshape_rater(kva_normalized, "KVA")
ly_long <- reshape_rater(ly_normalized, "LY")
wagner_long <- reshape_rater(wagner_normalized, "Wagner")
#tail(latchman_long)

```

```

all_ratings <- bind_rows(latchman_long, kva_long, ly_long, wagner_long)
all_ratings

```

```

## # A tibble: 1,440 x 4
##   Criterion Question Score Rater
##   <int> <chr> <chr> <chr>
## 1      1 Q1      N   Latchman
## 2      1 Q2      Y   Latchman
## 3      1 Q3      Y   Latchman
## 4      1 Q4      N   Latchman
## 5      1 Q5      N   Latchman
## 6      1 Q6      Y   Latchman
## 7      1 Q7      Y   Latchman
## 8      1 Q8      Y   Latchman
## 9      1 Q9      Y   Latchman
## 10     1 Q10     Y   Latchman
## # i 1,430 more rows

```

```

# Initialize results dataframe
kappa_results <- data.frame(
  Criterion = 1:12,
  Subjects = NA,
  Kappa = NA,
  z = NA,
  p_value = NA
)

for (i in 1:12) {

```

```

# Reshape data for this criterion
crit <- all_ratings %>%
  filter(Criterion == i) %>%
  pivot_wider(names_from = Rater, values_from = Score) %>%
  select(-Question)

# Drop Criterion column
crit_clean <- crit %>%
  select(-Criterion)

# Filter rows with at least 2 non-NA scores
crit_matrix <- as.matrix(crit_clean[rowSums(!is.na(crit_clean)) >= 2, ])

# Only run kappa if matrix is valid
if (nrow(crit_matrix) >= 2) {
  result <- tryCatch(
    kappam.light(crit_matrix),
    error = function(e) NULL
  )

  if (!is.null(result)) {
    kappa_results$Subjects[i] <- result$subjects
    kappa_results$Kappa[i] <- result$value
    kappa_results$z[i] <- result$statistic
    kappa_results$p_value[i] <- result$p.value
  }
}
}

print(kappa_results)

```

##	Criterion	Subjects	Kappa	z	p_value
## 1	1	30	0.246976365	0.1854160407	0.8529027
## 2	2	30	-0.036062378	-0.0004747116	0.9996212
## 3	3	27	0.421330140	0.0674526523	0.9462214
## 4	4	23	NaN	NaN	NaN
## 5	5	4	0.666666667	0.3666793988	0.7138582
## 6	6	25	NaN	NaN	NaN
## 7	7	28	0.825000000	0.0168402534	0.9865641
## 8	8	21	0.097725086	0.0262910017	0.9790252
## 9	9	26	0.679639581	0.1058405191	0.9157089
## 10	10	4	-0.222222222	-0.0594503949	0.9525934
## 11	11	26	NaN	NaN	NaN
## 12	12	26	0.008623853	0.0002015462	0.9998392

```

for (i in 1:12) {
  cat("\n=== Criterion", i, "===\n")

  df <- all_ratings %>%
    filter(Criterion == i) %>%
    pivot_wider(names_from = Rater, values_from = Score) %>%
    select(-Question, -Criterion)
}

```

```

df[df == "N/A"] <- NA
df_filtered <- df[rowSums(!is.na(df)) >= 2, ]

if (nrow(df_filtered) == 0) {
  cat("No usable rows after filtering.\n")
} else {
  print(table(apply(df_filtered, 1, paste, collapse = "-")))
}
}

```

```

##
## === Criterion 1 ===
##
## N-N-N-N N-Y-N-N N-Y-N-Y Y-Y-N-N Y-Y-N-Y Y-Y-Y-Y
##      1      2      5      1     10     11
##
## === Criterion 2 ===
##
## N-Y-Y-Y Y-N-Y-Y Y-Y-Y-N Y-Y-Y-Y
##      3      1      3     23
##
## === Criterion 3 ===
##
## N-N-N-Y N-NA-N-Y N-Y-N-N Y-NA-Y-Y Y-Y-N-Y Y-Y-Y-Y
##      1      1      3      2      3     20
##
## === Criterion 4 ===
##
## N-NA-N-Y N-Y-N-Y N-Y-Y-Y Y-NA-N-Y Y-NA-Y-N Y-NA-Y-Y Y-Y-N-Y Y-Y-Y-Y
##      1      1      1      2      2      2      4     17
##
## === Criterion 5 ===
##
## N-N-N-N N-Y-N-Y Y-Y-Y-Y
##      1      1      2
##
## === Criterion 6 ===
##
## Y-N-N-Y Y-NA-Y-NA Y-NA-Y-Y Y-Y-Y-Y
##      1      2      3     24
##
## === Criterion 7 ===
##
## N-N-N-N N-N-Y-N Y-NA-Y-Y Y-Y-Y-Y
##      1      1      2     26
##
## === Criterion 8 ===
##
## N-NA-N-N N-NA-Y-Y N-Y-N-N N-Y-Y-NA N-Y-Y-Y Y-NA-N-N Y-NA-N-Y Y-NA-Y-Y
##      1      1      1      1      1      1      1      1
## Y-Y-N-N Y-Y-N-Y Y-Y-Y-N Y-Y-Y-NA Y-Y-Y-Y
##      3      5      1      1     10
##

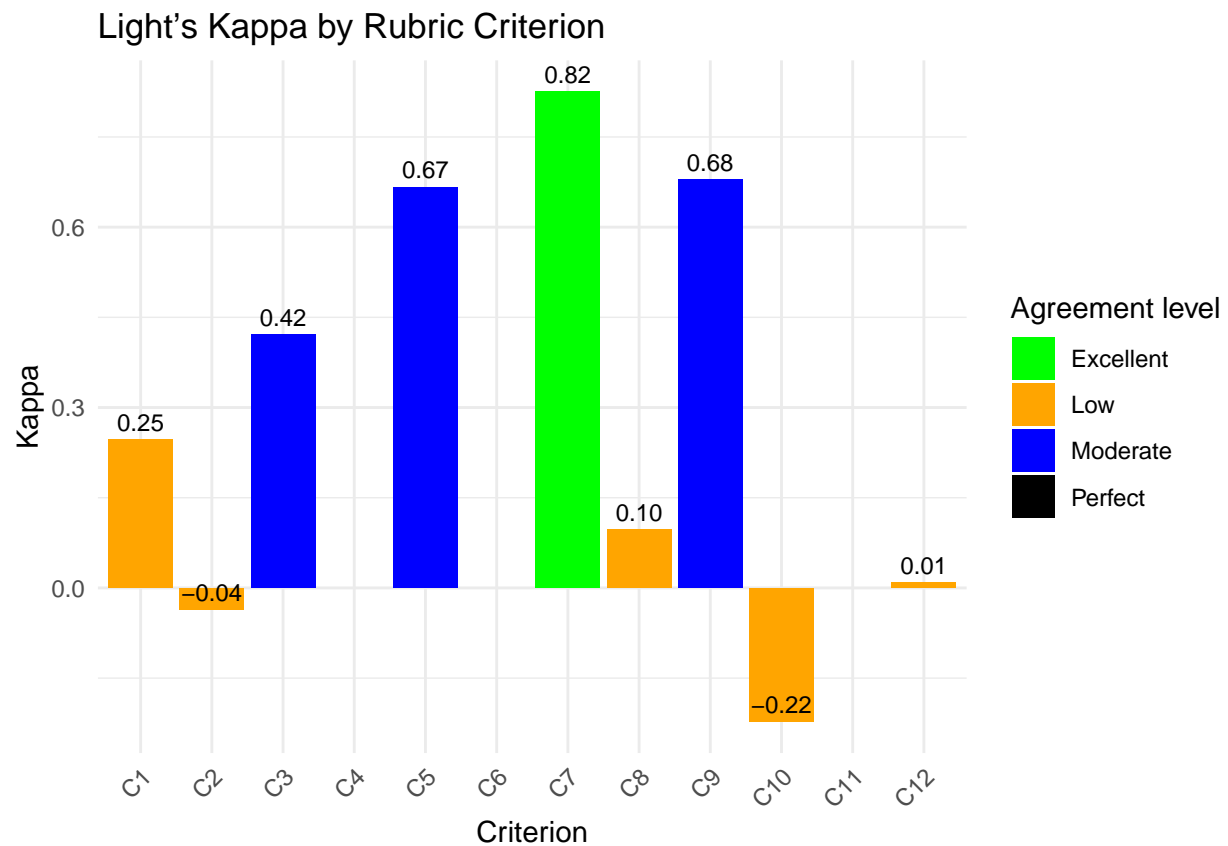
```

```
## === Criterion 9 ===
##
## N-N-N-N N-N-N-Y Y-Y-N-Y Y-Y-Y-N Y-Y-Y-NA Y-Y-Y-Y
##      2      1      2      1      4      20
##
## === Criterion 10 ===
##
## N-NA-N-Y N-NA-Y-Y N-Y-Y-Y NA-N-N-N NA-NA-Y-Y NA-Y-Y-Y Y-NA-N-Y Y-NA-Y-Y
##      1      1      1      1      6      2      3      2
## Y-Y-N-Y Y-Y-Y-N
##      2      1
##
## === Criterion 11 ===
##
## Y-NA-Y-NA Y-Y-N-Y Y-Y-Y-N Y-Y-Y-NA Y-Y-Y-Y
##      2      2      1      2      23
##
## === Criterion 12 ===
##
## N-Y-N-Y Y-NA-Y-NA Y-Y-N-Y Y-Y-Y-N Y-Y-Y-NA Y-Y-Y-Y
##      1      2      7      1      2      17
```

```
plot_data <- kappa_results %>%
  mutate(
    Criterion = paste0("C", Criterion),
    Criterion = factor(Criterion, levels = paste0("C", 1:12)),
    Label = ifelse(is.nan(Kappa), "Perfect\nagreement", sprintf("%.2f", round(Kappa, 2))),
    Fill = case_when(
      is.nan(Kappa) ~ "Perfect",
      Kappa >= 0.75 ~ "Excellent",
      Kappa >= 0.4 ~ "Moderate",
      !is.na(Kappa) ~ "Low",
      TRUE ~ "Missing"
    )
  )

ggplot(plot_data, aes(x = Criterion, y = Kappa, fill = Fill)) +
  geom_col(na.rm = TRUE) +
  geom_text(aes(label = Label), vjust = -0.5, size = 3) +
  scale_fill_manual(
    values = c(
      "Perfect" = "black",
      "Excellent" = "green",
      "Moderate" = "blue",
      "Low" = "orange",
      "Missing" = "gray"
    )
  ) +
  labs(
    title = "Light's Kappa by Rubric Criterion",
    y = "Kappa",
    x = "Criterion",
    fill = "Agreement level"
  ) +
```

```
theme_minimal() +
theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



## part 2: One Kappa per Question (All Criteria Combined)

```
# test on q1
q1_data <- all_ratings %>%
  filter(Question == "Q1") %>%
  pivot_wider(names_from = Rater, values_from = Score) %>%
  select(-Question, -Criterion)

q1_data[q1_data == "N/A"] <- NA

# Keep only criteria (rows) with at least 2 valid scores
q1_filtered <- q1_data[rowSums(!is.na(q1_data)) >= 2, ]

if (nrow(q1_filtered) >= 2) {
  q1_matrix <- as.matrix(q1_filtered)
  kappa_q1 <- kappam.light(q1_matrix)
  print(kappa_q1)
} else {
  print("Not enough usable criteria to compute kappa for Question 1.")
}
```

```

## Light's Kappa for m Raters
##
## Subjects = 10
## Raters = 4
## Kappa = 0.572
##
## z = 0.189
## p-value = 0.85

light_kappa_by_question <- data.frame(
  Question = character(),
  Subjects = integer(),
  Raters = integer(),
  Kappa = numeric(),
  Z = numeric(),
  P_value = numeric(),
  stringsAsFactors = FALSE
)

for (q in paste0("Q", 1:30)) {
  q_data <- all_ratings %>%
    filter(Question == q) %>%
    pivot_wider(names_from = Rater, values_from = Score) %>%
    select(-Question, -Criterion)

  q_data[q_data == "N/A"] <- NA

  q_filtered <- q_data[rowSums(!is.na(q_data)) >= 2, ]

  if (nrow(q_filtered) >= 2) {
    q_matrix <- as.matrix(q_filtered)
    result <- tryCatch({
      kappa <- kappam.light(q_matrix)
      data.frame(
        Question = q,
        Subjects = kappa$subjects,
        Raters = kappa$raters,
        Kappa = kappa$value,
        Z = kappa$statistic,
        P_value = kappa$p.value
      )
    }, error = function(e) {
      data.frame(
        Question = q,
        Subjects = nrow(q_filtered),
        Raters = ncol(q_filtered),
        Kappa = NA,
        Z = NA,
        P_value = NA
      )
    })
  } else {
    result <- data.frame(
      Question = q,

```

```

    Subjects = nrow(q_filtered),
    Raters = ifelse(nrow(q_data) > 0, ncol(q_data), NA),
    Kappa = NA,
    Z = NA,
    P_value = NA
  )
}

light_kappa_by_question <- rbind(light_kappa_by_question, result)
}

print(light_kappa_by_question)

```

##	Question	Subjects	Raters	Kappa	Z	P_value
## 1	Q1	10	4	0.57236842	0.189019680	0.8500774
## 2	Q2	9	4	NaN	NaN	NaN
## 3	Q3	10	4	0.12301587	0.035215915	0.9719076
## 4	Q4	9	4	0.14835874	0.043249103	0.9655030
## 5	Q5	10	4	0.04715219	0.007644911	0.9939003
## 6	Q6	10	4	NaN	NaN	NaN
## 7	Q7	6	4	1.00000000	0.128778425	0.8975330
## 8	Q8	10	4	NA	NA	NA
## 9	Q9	11	4	NaN	NaN	NaN
## 10	Q10	11	4	NA	NA	NA
## 11	Q11	10	4	0.20256410	0.040081440	0.9680282
## 12	Q12	9	4	NaN	NaN	NaN
## 13	Q13	11	4	NA	NA	NA
## 14	Q14	9	4	0.50000000	0.011947288	0.9904677
## 15	Q15	10	4	NaN	NaN	NaN
## 16	Q16	7	4	0.26819990	0.092573330	0.9262425
## 17	Q17	8	4	NaN	NaN	NaN
## 18	Q18	8	4	NaN	NaN	NaN
## 19	Q19	11	4	NA	NA	NA
## 20	Q20	11	4	NA	NA	NA
## 21	Q21	10	4	0.69696970	0.416892317	0.6767572
## 22	Q22	10	4	NaN	NaN	NaN
## 23	Q23	10	4	NaN	NaN	NaN
## 24	Q24	9	4	0.26242236	0.023858471	0.9809655
## 25	Q25	11	4	NA	NA	NA
## 26	Q26	10	4	0.30392157	0.018603484	0.9851574
## 27	Q27	11	4	NaN	NaN	NaN
## 28	Q28	6	4	0.10000000	0.005978606	0.9952298
## 29	Q29	9	4	0.73913043	0.114587357	0.9087722
## 30	Q30	10	4	0.40000000	0.083839999	0.9331836