Measurement checks — HPT (Czech data)

Reliability, dimensionality, presentism-contextualization contrast, and ICCs

HPT and Extremism project

2025-12-12

Contents

1 What this document does

This report checks whether our **Historical Perspective-Taking (HPT)** instrument behaves well **before** we run any hypothesis tests.

We do four things:

1. **Reliability:** Are the HPT subscales internally consistent? We report **Cronbach's alpha** (α) and **McDonald's omega** (ω) for **POP**, **ROA**, **CONT** (three items each; 1–4).
2. **Dimensionality (CFA/EFA):** Does the **factor structure** match prior research (e.g., **POP+CONT** vs **ROA**, or three correlated factors)?
3. **Presentism–contextualization contrast:** Do **POP** (presentist) and **CONT** (contextualization) show the expected contrast?
4. **Clustering (ICCs):** Are scores clustered by **school** and **class-within-school**?

   Composite scores used (for descriptives/ICCs and later files): - **POP_rev = 5 − POP_raw** (so higher = more contextualized). - **HPT_CTX6 = mean(POP_rev, CONT)** (default composite). - **HPT_TOT9 = mean(POP_rev, CONT, ROA)** (includes ROA; robustness).

2 Setup and data loading

```
options(width = 120)

# Data handling & plots
library(tidyverse)

# Psychometrics
library(psych)        # alpha, omega, polychoric, EFA helpers
library(lavaan)       # CFA
library(semTools)     # model comparisons & extras
```

```r
# Multilevel ICCs
library(lme4)
library(performance)

# Tables
library(knitr)

# Make kableExtra use longtable/booktabs and avoid loading tabu
options(kableExtra.latex.load_packages = FALSE)
library(kableExtra)

# Load the dataset created in 00_data-preparation
load("normalised_responses.RData")
stopifnot(exists("normalised_responses"))
dat <- normalised_responses

# Ensure clustering identifiers and unique class id
stopifnot(all(c("school_id","class_label") %in% names(dat)))
dat <- dat %>%
  mutate(
    school_id   = as.factor(school_id),
    class_label = as.factor(class_label),
    class_id    = interaction(school_id, class_label, drop = TRUE)
  )

# POP reversed item-wise and subscale helper
POP_rev_items <- paste0("POP", 1:3)

# Reverse POP items (1-4)
dat <- dat %>%
  mutate(
    across(
      all_of(POP_rev_items),
      ~ 5 - suppressWarnings(as.numeric(.)),
      .names = "{.col}_rev"     # <<< THIS was the culprit
    )
  ) %>%
  mutate(
    HPT_POP_raw = rowMeans(across(all_of(POP_rev_items)), na.rm = TRUE),
    HPT_POP_rev = rowMeans(across(all_of(paste0(POP_rev_items, "_rev"))), na.rm = TRUE),
    HPT_CONT    = rowMeans(across(CONT1:CONT3), na.rm = TRUE),
    HPT_ROA     = rowMeans(across(ROA1:ROA3),   na.rm = TRUE),
```

```r
    HPT_CTX6     = rowMeans(cbind(HPT_POP_rev, HPT_CONT), na.rm = TRUE),
    HPT_TOT9     = rowMeans(cbind(HPT_POP_rev, HPT_CONT, HPT_ROA), na.rm = TRUE)
  )

print_tbl <- function(df, caption, digits = 3, escape = TRUE) {
  kbl(df, booktabs = TRUE, longtable = TRUE, caption = caption, digits = digits, escape = escape) |>
    kable_styling(full_width = FALSE, latex_options = c("hold_position"))
}
```

We verify that **HPT items** and **class labels** exist. If something is missing, we stop with a clear message.

```r
## -- check-columns ----------------------------
hpt_cols <- c(paste0("POP", 1:3), paste0("ROA", 1:3), paste0("CONT", 1:3))
need   <- c(hpt_cols, "class_label", "school_id")
miss   <- setdiff(need, names(dat))
if (length(miss)) stop("Missing variables: ", paste(miss, collapse = ", "))

# Keep only rows that are COMPLETE on all HPT items AND have class_label and school_id
keep <- complete.cases(dat[, hpt_cols]) & !is.na(dat$class_label) & !is.na(dat$school_id)

analysis_df <- dat[keep, c(hpt_cols, "class_label", "school_id", "class_id")] |>
  as_tibble()

nrow_all   <- nrow(dat)
nrow_keep  <- nrow(analysis_df)
cat("Rows in full data: ", nrow_all,  "\n",
    "Rows kept (complete HPT + class_label + school_id): ", nrow_keep, "\n", sep = "")
```

```
## Rows in full data: 234
## Rows kept (complete HPT + class_label + school_id): 221
```

3 Step 1 — Descriptives and scale construction

Why: Simple summaries catch obvious data problems and help readers develop intuition.

```r
hpt_items <- analysis_df %>% select(all_of(hpt_cols))  # 9 HPT items

# Subscales and composites with POP reversed for composites
hpt_scores <- hpt_items %>%
  mutate(
```

```r
    POP_raw = rowMeans(select(., starts_with("POP")),  na.rm = TRUE),
    ROA     = rowMeans(select(., starts_with("ROA")),  na.rm = TRUE),
    CONT    = rowMeans(select(., starts_with("CONT")), na.rm = TRUE)
  ) %>%
  mutate(
    POP_rev   = 5 - POP_raw,
    HPT_CTX6  = rowMeans(cbind(POP_rev, CONT), na.rm = TRUE),
    HPT_TOT9  = rowMeans(cbind(POP_rev, CONT, ROA), na.rm = TRUE)
  )

summary(select(hpt_scores, POP_raw, POP_rev, ROA, CONT, HPT_CTX6, HPT_TOT9))
```

```
##     POP_raw          POP_rev            ROA              CONT           HPT_CTX6         HPT_TOT9
##  Min.   :1.000   Min.   :1.333   Min.   :1.000   Min.   :1.000   Min.   :1.333   Min.   :1.667
##  1st Qu.:1.333   1st Qu.:2.333   1st Qu.:2.333   1st Qu.:2.333   1st Qu.:2.500   1st Qu.:2.444
##  Median :2.000   Median :3.000   Median :3.000   Median :2.667   Median :2.833   Median :2.889
##  Mean   :2.017   Mean   :2.983   Mean   :2.804   Mean   :2.735   Mean   :2.859   Mean   :2.841
##  3rd Qu.:2.667   3rd Qu.:3.667   3rd Qu.:3.333   3rd Qu.:3.333   3rd Qu.:3.333   3rd Qu.:3.222
##  Max.   :3.667   Max.   :4.000   Max.   :4.000   Max.   :4.000   Max.   :4.000   Max.   :3.889
```

```r
cor(select(hpt_scores, POP_raw, ROA, CONT, HPT_CTX6, HPT_TOT9), use = "pairwise.complete.obs")
```

```
##             POP_raw        ROA       CONT   HPT_CTX6   HPT_TOT9
## POP_raw   1.0000000 -0.1515670 -0.3568641 -0.8024568 -0.6851161
## ROA      -0.1515670  1.0000000  0.3678874  0.3220708  0.6827642
## CONT     -0.3568641  0.3678874  1.0000000  0.8437887  0.8109413
## HPT_CTX6 -0.8024568  0.3220708  0.8437887  1.0000000  0.9116056
## HPT_TOT9 -0.6851161  0.6827642  0.8109413  0.9116056  1.0000000
```

4 Step 2 — Reliability: α and ω for POP–ROA–CONT

```r
alpha_poly <- function(x) {
  pc <- psych::polychoric(x)$rho
  psych::alpha(pc, n.obs = nrow(x))
}
omega_poly <- function(x) {
  pc <- psych::polychoric(x)$rho
  psych::omega(pc, n.obs = nrow(x), nfactors = 1, plot = FALSE)
```

```
}

subsets <- list(
  POP  = hpt_items %>% select(starts_with("POP")),
  ROA  = hpt_items %>% select(starts_with("ROA")),
  CONT = hpt_items %>% select(starts_with("CONT"))
)

rel_table <- purrr::imap_dfr(subsets, function(df, nm){
  a_raw  <- psych::alpha(df)
  a_poly <- alpha_poly(df)
  om     <- omega_poly(df)
  tibble(
    scale = nm,
    k_items = ncol(df),
    alpha_raw  = unname(a_raw$total$raw_alpha),
    alpha_poly = unname(a_poly$total$raw_alpha),
    omega_total = unname(om$omega.tot),
    omega_hier  = unname(om$omega.h)
  )
})
```

```
## Loading required namespace: GPArotation
```

```
## Omega_h for 1 factor is not meaningful, just omega_t
## Omega_h for 1 factor is not meaningful, just omega_t
## Omega_h for 1 factor is not meaningful, just omega_t
```

```
print_tbl(rel_table, digits = 3, caption = "Reliability of HPT subscales (alpha and omega).")
```

Table 1: Reliability of HPT subscales (alpha and omega).

scale	k_items	alpha_raw	alpha_poly	omega_total
POP	3	0.479	0.535	0.560
ROA	3	0.432	0.453	0.538
CONT	3	0.635	0.690	0.691

5 Step 3 — Dimensionality (CFA/EFA)

```
hpt_ord <- hpt_items  # treat items as ordered

m1_2factor <- '
F1 =~ POP1 + POP2 + POP3 + CONT1 + CONT2 + CONT3
F2 =~ ROA1 + ROA2 + ROA3
F1 ~~ F2
'
m2_3factor <- '
POP  =~ POP1 + POP2 + POP3
CONT =~ CONT1 + CONT2 + CONT3
ROA  =~ ROA1 + ROA2 + ROA3
POP ~~ CONT + ROA
CONT ~~ ROA
'
m3_1factor <- '
G =~ POP1 + POP2 + POP3 + ROA1 + ROA2 + ROA3 + CONT1 + CONT2 + CONT3
'

fit_2 <- cfa(m1_2factor, data = hpt_ord, ordered = hpt_cols, estimator = "WLSMV")
fit_3 <- cfa(m2_3factor, data = hpt_ord, ordered = hpt_cols, estimator = "WLSMV")
fit_1 <- cfa(m3_1factor, data = hpt_ord, ordered = hpt_cols, estimator = "WLSMV")

# Compare fits
semTools::compareFit(fit_2, fit_3, fit_1)
```

```
## The following lavaan models were compared:
##     fit_3
##     fit_2
##     fit_1
## To view results, assign the compareFit() output to an object and  use the summary() method; see the class?FitDiff help page.
```

```
report_fit <- function(fit) {
  list(
    indices = fitMeasures(fit, c("cfi","tli","rmsea","rmsea.ci.lower","rmsea.ci.upper","srmr")),
    loadings = standardizedSolution(fit) %>% as_tibble() %>% filter(op == "=~")
  )
}
```

```r
cfa_summary <- list(
  `2-factor (POP+CONT vs ROA)` = report_fit(fit_2),
  `3-factor (POP/CONT/ROA)`    = report_fit(fit_3),
  `1-factor (general)`         = report_fit(fit_1)
)

purrr::iwalk(cfa_summary, function(x, nm){
  cat("\n###", nm, "\n")
  print(x$indices)
  print(kable(x$loadings, digits = 3))
})
```

```
##
## ### 2-factor (POP+CONT vs ROA)
##           cfi            tli          rmsea rmsea.ci.lower rmsea.ci.upper           srmr
##         0.971          0.960          0.048          0.004          0.077          0.066
##
##
## |lhs |op |rhs   | est.std|    se|       z| pvalue| ci.lower| ci.upper|
## |:---|:--|:-----|-------:|-----:|-------:|------:|--------:|--------:|
## |F1  |=~ |POP1  |   0.571| 0.069|   8.264|  0.000|    0.435|    0.706|
## |F1  |=~ |POP2  |   0.230| 0.072|   3.186|  0.001|    0.089|    0.372|
## |F1  |=~ |POP3  |   0.413| 0.062|   6.662|  0.000|    0.291|    0.534|
## |F1  |=~ |CONT1 |  -0.687| 0.061| -11.290|  0.000|   -0.806|   -0.567|
## |F1  |=~ |CONT2 |  -0.592| 0.064|  -9.218|  0.000|   -0.718|   -0.466|
## |F1  |=~ |CONT3 |  -0.620| 0.060| -10.290|  0.000|   -0.738|   -0.502|
## |F2  |=~ |ROA1  |   0.611| 0.088|   6.955|  0.000|    0.439|    0.783|
## |F2  |=~ |ROA2  |   0.187| 0.089|   2.099|  0.036|    0.012|    0.361|
## |F2  |=~ |ROA3  |   0.615| 0.084|   7.341|  0.000|    0.451|    0.780|
##
## ### 3-factor (POP/CONT/ROA)
##           cfi            tli          rmsea rmsea.ci.lower rmsea.ci.upper           srmr
##         0.997          0.995          0.016          0.000          0.058          0.054
##
##
## |lhs  |op |rhs   | est.std|    se|      z| pvalue| ci.lower| ci.upper|
## |:----|:--|:-----|-------:|-----:|------:|------:|--------:|--------:|
## |POP  |=~ |POP1  |   0.759| 0.094|  8.061|   0.00|    0.574|    0.943|
## |POP  |=~ |POP2  |   0.297| 0.080|  3.711|   0.00|    0.140|    0.453|
## |POP  |=~ |POP3  |   0.510| 0.066|  7.689|   0.00|    0.380|    0.640|
## |CONT |=~ |CONT1 |   0.709| 0.062| 11.487|   0.00|    0.588|    0.830|
## |CONT |=~ |CONT2 |   0.607| 0.065|  9.328|   0.00|    0.479|    0.734|
```

```
## |CONT |=~ |CONT3 |   0.641| 0.061| 10.486|   0.00|    0.521|    0.761|
## |ROA  |=~ |ROA1  |   0.613| 0.087|  7.072|   0.00|    0.443|    0.782|
## |ROA  |=~ |ROA2  |   0.192| 0.089|  2.169|   0.03|    0.018|    0.366|
## |ROA  |=~ |ROA3  |   0.612| 0.083|  7.399|   0.00|    0.450|    0.774|
##
## ### 1-factor (general)
##            cfi            tli          rmsea rmsea.ci.lower rmsea.ci.upper           srmr
##          0.952          0.935          0.061          0.032          0.088          0.073
##
##
## |lhs |op |rhs   | est.std|    se|       z| pvalue| ci.lower| ci.upper|
## |:---|:--|:-----|-------:|-----:|-------:|------:|--------:|--------:|
## |G   |=~ |POP1  |   0.563| 0.069|   8.143|  0.000|    0.427|    0.698|
## |G   |=~ |POP2  |   0.219| 0.072|   3.048|  0.002|    0.078|    0.360|
## |G   |=~ |POP3  |   0.403| 0.062|   6.501|  0.000|    0.282|    0.525|
## |G   |=~ |ROA1  |  -0.458| 0.070|  -6.511|  0.000|   -0.596|   -0.320|
## |G   |=~ |ROA2  |  -0.135| 0.076|  -1.784|  0.074|   -0.284|    0.013|
## |G   |=~ |ROA3  |  -0.465| 0.066|  -7.021|  0.000|   -0.594|   -0.335|
## |G   |=~ |CONT1 |  -0.676| 0.059| -11.393|  0.000|   -0.792|   -0.560|
## |G   |=~ |CONT2 |  -0.585| 0.063|  -9.233|  0.000|   -0.709|   -0.461|
## |G   |=~ |CONT3 |  -0.608| 0.058| -10.400|  0.000|   -0.723|   -0.494|
```

5.0.1 Optional: EFA (polychoric)

```r
pc <- psych::polychoric(hpt_ord)$rho
efa2 <- psych::fa(pc, nfactors = 2, fm = "pa", rotate = "oblimin")
efa3 <- psych::fa(pc, nfactors = 3, fm = "pa", rotate = "oblimin")

cat("\nEFA (2 factors):\n")
```

```
##
## EFA (2 factors):
```

```r
print(efa2$loadings, cutoff = 0.25)
```

```
##
## Loadings:
##      PA1    PA2
## POP1 -0.442  0.266
## POP2          0.534
```

```
## POP3          0.472
## ROA1   0.569
## ROA2   0.292  0.301
## ROA3   0.554
## CONT1  0.655
## CONT2  0.524
## CONT3  0.532
##
##                  PA1   PA2
## SS loadings    1.934 0.763
## Proportion Var 0.215 0.085
## Cumulative Var 0.215 0.300
```

```
cat("\nEFA (3 factors):\n")
```

```
##
## EFA (3 factors):
```

```
print(efa3$loadings, cutoff = 0.25)
```

```
##
## Loadings:
##       PA1    PA2    PA3
## POP1          0.400
## POP2          0.362  0.284
## POP3          0.744
## ROA1                 0.479
## ROA2                 0.348
## ROA3                 0.556
## CONT1  0.479
## CONT2  0.521
## CONT3  0.765
##
##                  PA1   PA2   PA3
## SS loadings    1.189 0.888 0.861
## Proportion Var 0.132 0.099 0.096
## Cumulative Var 0.132 0.231 0.326
```

6 Step 4 — Presentism–contextualization contrast (POP vs CONT)

```
contrast_tbl <- hpt_scores %>%
  summarise(
    mean_POP    = mean(POP_raw,  na.rm = TRUE),  sd_POP    = sd(POP_raw,  na.rm = TRUE),
    mean_CONT   = mean(CONT,     na.rm = TRUE),  sd_CONT   = sd(CONT,     na.rm = TRUE),
    r_POP_CONT = cor(POP_raw, CONT, use = "pairwise.complete.obs")
  )

print_tbl(contrast_tbl, digits = 3, caption = "POP (raw) vs CONT: means, SDs, and correlation.")
```

Table 2: POP (raw) vs CONT: means, SDs, and correlation.

mean_POP	sd_POP	mean_CONT	sd_CONT	r_POP_CONT
2.017	0.656	2.735	0.73	-0.357

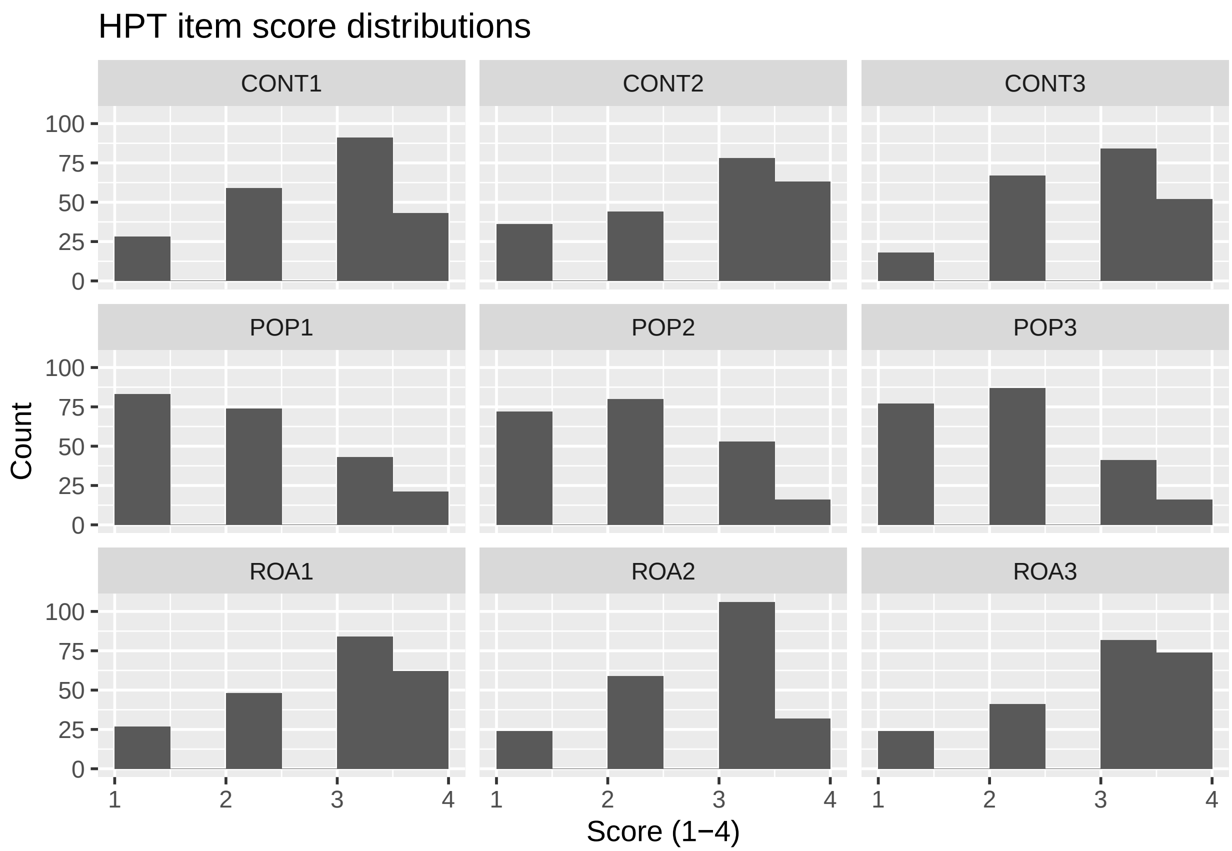
```
t_test <- t.test(hpt_scores$POP_raw, hpt_scores$CONT, paired = TRUE)
t_test
```

```
##
##  Paired t-test
##
## data:  hpt_scores$POP_raw and hpt_scores$CONT
## t = -9.345, df = 220, p-value < 2.2e-16
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  -0.8693599 -0.5665376
## sample estimates:
## mean difference
##      -0.7179487
```

7 Step 5 — Distribution checks

```
# Item distributions
long_items <- hpt_items %>%
  pivot_longer(cols = everything(), names_to = "item", values_to = "score")
```

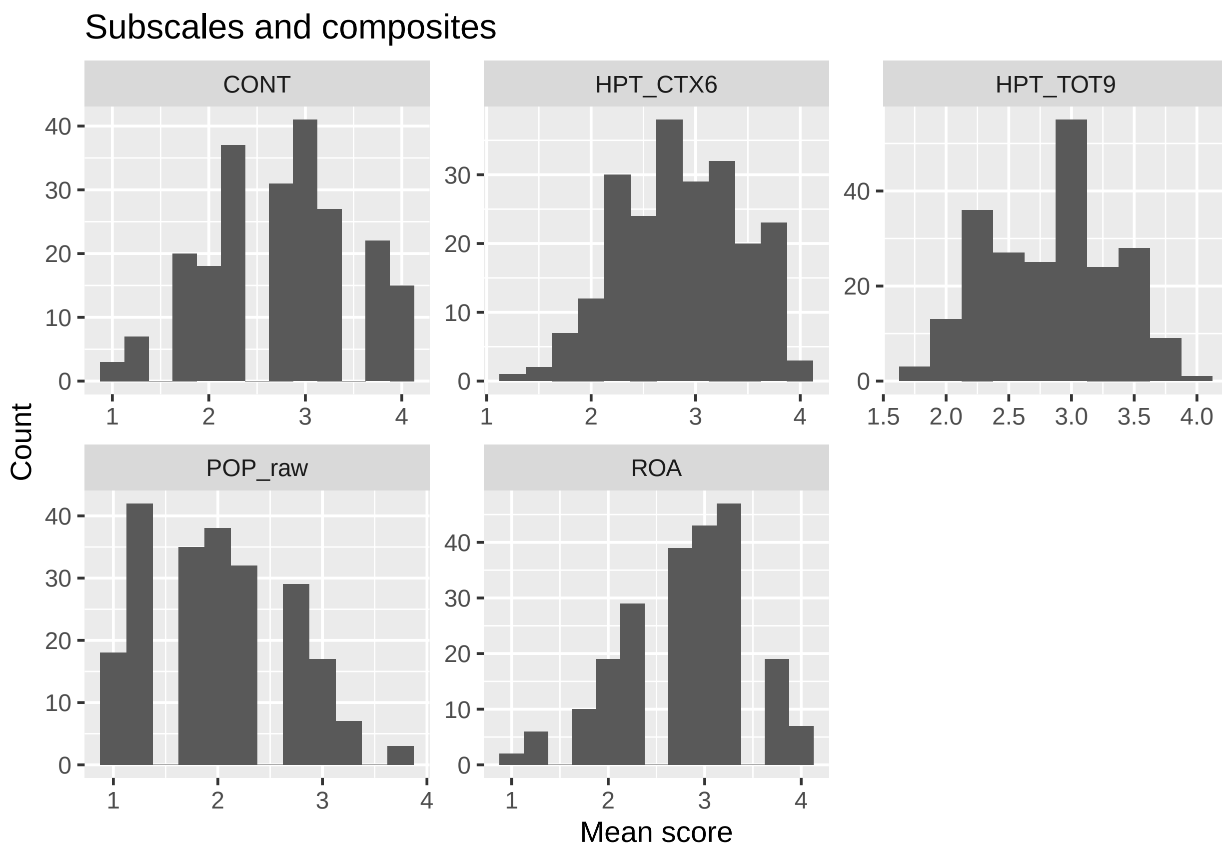
```r
ggplot(long_items, aes(score)) +
  geom_histogram(binwidth = 0.5, boundary = 0, closed = "left") +
  facet_wrap(~ item, ncol = 3) +
  labs(title = "HPT item score distributions", x = "Score (1-4)", y = "Count")
```




```r
# Scale/composite distributions
long_scales <- hpt_scores %>%
  select(POP_raw, ROA, CONT, HPT_CTX6, HPT_TOT9) %>%
  pivot_longer(everything(), names_to = "scale", values_to = "score")

ggplot(long_scales, aes(x = score)) +
  geom_histogram(binwidth = 0.25) +
  facet_wrap(~ scale, scales = "free") +
```

```r
  labs(title = "Subscales and composites", x = "Mean score", y = "Count")
```



8 Step 6 — Class-level ICCs (multilevel warranted?)

```r
# Build the DV frame from analysis_df (items) and compute subscales/composites here
icc_data <- analysis_df %>%
  mutate(
    POP_raw = rowMeans(select(., starts_with("POP")),  na.rm = TRUE),
    ROA     = rowMeans(select(., starts_with("ROA")),  na.rm = TRUE),
    CONT    = rowMeans(select(., starts_with("CONT")), na.rm = TRUE)
  ) %>%
```

```r
  mutate(
    POP_rev  = 5 - POP_raw,
    HPT_CTX6 = rowMeans(cbind(POP_rev, CONT), na.rm = TRUE),
    HPT_TOT9 = rowMeans(cbind(POP_rev, CONT, ROA), na.rm = TRUE)
  ) %>%
  select(school_id, class_label, class_id, POP_raw, ROA, CONT, HPT_CTX6, HPT_TOT9)

mk_icc_3 <- function(dv){
  f <- reformulate("1 + (1|school_id) + (1|school_id:class_label)", response = dv)
  fit <- lmer(f, data = icc_data, REML = TRUE)

  vc <- as.data.frame(VarCorr(fit))
  v_school <- vc$vcov[vc$grp == "school_id"];              v_school <- if (length(v_school)) v_school[1] else 0
  v_classW <- vc$vcov[vc$grp == "school_id:class_label"]; v_classW <- if (length(v_classW)) v_classW[1] else 0
  v_resid  <- vc$vcov[vc$grp == "Residual"][1]
  v_tot    <- v_school + v_classW + v_resid

  tibble(
    DV = dv,
    ICC_school = v_school / v_tot,
    ICC_class_within_school = v_classW / v_tot,
    ICC_total_cluster = (v_school + v_classW) / v_tot,
    singular = isSingular(fit)
  )
}

icc_tbl <- purrr::map_dfr(c("HPT_CTX6","HPT_TOT9","POP_raw","ROA","CONT"), mk_icc_3)
```

```
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
```

```r
print_tbl(icc_tbl, digits = 3, caption = "ICCs: school and class-within-school (HPT composites and subscales).")
```

Table 3: ICCs: school and class-within-school (HPT composites and subscales).

DV	ICC_school	ICC_class_within_school	ICC_total_cluster	singular
HPT_CTX6	0.050	0.000	0.050	TRUE
HPT_TOT9	0.081	0.000	0.081	TRUE
POP_raw	0.036	0.000	0.036	TRUE

ROA	0.037	0.003	0.040	FALSE
CONT	0.040	0.012	0.053	FALSE

9 Step 7 — Knowledge mini-test (KN1–KN6)

```
kn_cols <- paste0("KN", 1:6)
has_kn  <- all(kn_cols %in% names(dat))

if (!has_kn) {
  cat("\n**Knowledge section skipped:** KN1-KN6 not found in data.\n")
} else {
  kn_items <- dat[keep, kn_cols]  # align to analysis_df rows via 'keep'
  # Basic sanity: coerce to numeric 0/1
  kn_items <- kn_items %>% mutate(across(everything(), ~ as.numeric(.)))

  # Total score, difficulty (p), discrimination (point-biserial)
  kn_total <- rowSums(kn_items, na.rm = TRUE)

  item_stats <- tibble(
    item = kn_cols,
    difficulty_p = sapply(kn_items, function(x) mean(x, na.rm = TRUE)),
    discr_pb = sapply(kn_items, function(x) cor(x, kn_total - x, use = "pairwise.complete.obs"))
  )

  # KR-20 (alpha on dichotomous items)
  kn_alpha <- psych::alpha(kn_items)

  print_tbl(item_stats, digits = 3, caption = "KN items: difficulty (p) and point-biserial discrimination.")

  print_tbl(tibble(
    k_items = ncol(kn_items),
    total_mean = mean(kn_total, na.rm = TRUE),
    total_sd   = sd(kn_total, na.rm = TRUE),
    alpha_KR20 = unname(kn_alpha$total$raw_alpha)
  ), digits = 3, caption = "KN total: summary and KR-20 (alpha for dichotomous items).")
}
```

Table 4: KN total: summary and KR-20 (alpha for dichotomous items).

k_items	total_mean	total_sd	alpha_KR20
6	3.131	1.656	0.567

10 Step 8 — Ideology batteries (FR-LF mini, KSA-3) and Social Desirability (SDR-5)

```
# Helper: reliability table for Likert batteries (polychoric + omega total)
alpha_poly_likert <- function(x) {
  pc <- psych::polychoric(x)$rho
  psych::alpha(pc, n.obs = nrow(x))
}
omega_total_poly_likert <- function(x) {
  pc <- psych::polychoric(x)$rho
  if (!all(eigen(pc, symmetric = TRUE)$values > 1e-6)) pc <- psych::cor.smooth(pc)
  suppressWarnings(psych::omega(pc, n.obs = nrow(x), nfactors = 1, plot = FALSE)$omega.tot)
}
```

10.1 FR-LF mini (RD1–RD3, NS1–NS3)

```
fr_cols <- c(paste0("RD", 1:3), paste0("NS", 1:3))
has_fr  <- all(fr_cols %in% names(dat))

if (!has_fr) {
  cat("\n**FR-LF mini section skipped:** RD1-3 and/or NS1-3 not found.\n")
} else {
  fr_df <- dat[keep, fr_cols] %>% as_tibble()
  RD <- fr_df %>% select(starts_with("RD"))
  NS <- fr_df %>% select(starts_with("NS"))

  fr_rel <- bind_rows(
    {
      a <- psych::alpha(RD); ap <- alpha_poly_likert(RD); wt <- omega_total_poly_likert(RD)
      tibble(scale = "FR-LF: RD", k_items = ncol(RD),
             alpha_raw = a$total$raw_alpha, alpha_poly = ap$total$raw_alpha, omega_total = wt)
    },
    {
```

```r
    a <- psych::alpha(NS); ap <- alpha_poly_likert(NS); wt <- omega_total_poly_likert(NS)
    tibble(scale = "FR-LF: NS", k_items = ncol(NS),
           alpha_raw = a$total$raw_alpha, alpha_poly = ap$total$raw_alpha, omega_total = wt)
  },
  {
    a <- psych::alpha(fr_df); ap <- alpha_poly_likert(fr_df); wt <- omega_total_poly_likert(fr_df)
    tibble(scale = "FR-LF: total (RD+NS)", k_items = ncol(fr_df),
           alpha_raw = a$total$raw_alpha, alpha_poly = ap$total$raw_alpha, omega_total = wt)
  }
 )

 print_tbl(fr_rel, digits = 3, caption = "FR-LF mini reliability (alpha, polychoric alpha, omega total).")

 # Optional CFA: 2 correlated factors (RD, NS), ordered WLSMV
 fr_model <- '
 RD =~ RD1 + RD2 + RD3
 NS =~ NS1 + NS2 + NS3
 RD ~~ NS
 '
 fr_fit <- try(lavaan::cfa(fr_model, data = fr_df, ordered = colnames(fr_df), estimator = "WLSMV"), silent = TRUE)
 if (!inherits(fr_fit, "try-error")) {
   print(fitMeasures(fr_fit, c("cfi","tli","rmsea","srmr")))
 } else {
   cat("\nFR-LF CFA skipped (model failed to converge).\n")
 }
}
```

```
## Omega_h for 1 factor is not meaningful, just omega_t
## Omega_h for 1 factor is not meaningful, just omega_t
## Omega_h for 1 factor is not meaningful, just omega_t
```

```
##   cfi   tli rmsea  srmr
## 0.984 0.970 0.063 0.054
```

10.2 KSA-3 (A1–A3, U1–U3, K1–K3)

```r
ksa_cols <- c(paste0("A",1:3), paste0("U",1:3), paste0("K",1:3))
has_ksa  <- all(ksa_cols %in% names(dat))
```

```r
if (!has_ksa) {
  cat("\n**KSA-3 section skipped:** A1-A3, U1-U3, and/or K1-K3 not found.\n")
} else {
  ksa_df <- dat[keep, ksa_cols] %>% as_tibble()
  A <- ksa_df %>% select(starts_with("A"))
  U <- ksa_df %>% select(starts_with("U"))
  K <- ksa_df %>% select(starts_with("K"))

  ksa_rel <- bind_rows(
    {
      a <- psych::alpha(A); ap <- alpha_poly_likert(A); wt <- omega_total_poly_likert(A)
      tibble(scale = "KSA-3: Aggression (A)", k_items = 3,
             alpha_raw = a$total$raw_alpha, alpha_poly = ap$total$raw_alpha, omega_total = wt)
    },
    {
      a <- psych::alpha(U); ap <- alpha_poly_likert(U); wt <- omega_total_poly_likert(U)
      tibble(scale = "KSA-3: Submission (U)", k_items = 3,
             alpha_raw = a$total$raw_alpha, alpha_poly = ap$total$raw_alpha, omega_total = wt)
    },
    {
      a <- psych::alpha(K); ap <- alpha_poly_likert(K); wt <- omega_total_poly_likert(K)
      tibble(scale = "KSA-3: Conventionalism (K)", k_items = 3,
             alpha_raw = a$total$raw_alpha, alpha_poly = ap$total$raw_alpha, omega_total = wt)
    },
    {
      a <- psych::alpha(ksa_df); ap <- alpha_poly_likert(ksa_df); wt <- omega_total_poly_likert(ksa_df)
      tibble(scale = "KSA-3: total", k_items = 9,
             alpha_raw = a$total$raw_alpha, alpha_poly = ap$total$raw_alpha, omega_total = wt)
    }
  )

  print_tbl(ksa_rel, digits = 3, caption = "KSA-3 reliability (alpha, polychoric alpha, omega total).")

  # Optional CFA: 3 correlated factors (A, U, K)
  ksa_model <- '
  A =~ A1 + A2 + A3
  U =~ U1 + U2 + U3
  K =~ K1 + K2 + K3
  A ~~ U + K
  U ~~ K
  '
  ksa_fit <- try(lavaan::cfa(ksa_model, data = ksa_df, ordered = colnames(ksa_df), estimator = "WLSMV"), silent = TRUE)
```

```r
  if (!inherits(ksa_fit, "try-error")) {
    print(fitMeasures(ksa_fit, c("cfi","tli","rmsea","srmr")))
  } else {
    cat("\nKSA-3 CFA skipped (model failed to converge).\n")
  }
}
```

```
## Omega_h for 1 factor is not meaningful, just omega_t
## Omega_h for 1 factor is not meaningful, just omega_t
## Omega_h for 1 factor is not meaningful, just omega_t
## Omega_h for 1 factor is not meaningful, just omega_t
```

```
##   cfi   tli rmsea  srmr
## 0.971 0.957 0.065 0.065
```

10.3 SDR-5 (SDR1–SDR5)

```r
sdr_cols <- paste0("SDR", 1:5)
has_sdr  <- all(sdr_cols %in% names(dat))

if (!has_sdr) {
  cat("\n**SDR-5 section skipped:** SDR1-SDR5 not found.\n")
} else {
  sdr_df <- dat[keep, sdr_cols] %>% as_tibble()
  a_sdr  <- psych::alpha(sdr_df)
  ap_sdr <- alpha_poly_likert(sdr_df)
  wt_sdr <- omega_total_poly_likert(sdr_df)

  print_tbl(tibble(
    scale = "SDR-5",
    k_items = 5,
    alpha_raw = a_sdr$total$raw_alpha,
    alpha_poly = ap_sdr$total$raw_alpha,
    omega_total = wt_sdr
  ), digits = 3, caption = "SDR-5 reliability (alpha, polychoric alpha, omega total).")

  # Optional CFA: 1 factor
  sdr_model <- 'SDR =~ SDR1 + SDR2 + SDR3 + SDR4 + SDR5'
  sdr_fit <- try(lavaan::cfa(sdr_model, data = sdr_df, ordered = colnames(sdr_df), estimator = "WLSMV"), silent = TRUE)
```

```
  if (!inherits(sdr_fit, "try-error")) {
    print(fitMeasures(sdr_fit, c("cfi","tli","rmsea","srmr")))
  } else {
    cat("\nSDR-5 CFA skipped (model failed to converge).\n")
  }
}
```

```
## Some items ( SDR1 SDR5 ) were negatively correlated with the first principal component and
## probably should be reversed.
## To do this, run the function again with the 'check.keys=TRUE' option

## Some items ( SDR1 SDR5 ) were negatively correlated with the first principal component and
## probably should be reversed.
## To do this, run the function again with the 'check.keys=TRUE' option

## Omega_h for 1 factor is not meaningful, just omega_t

##   cfi   tli rmsea  srmr
## 0.776 0.552 0.187 0.105
```

11 Step 9 — Cross-construct correlations (HPT, KN, FR-LF, KSA-3, SDR-5)

```
# Build scale scores that exist in your data (gracefully skipping any missing block)
scales_list <- list(
  HPT_CTX6  = hpt_scores$HPT_CTX6,
  HPT_TOT9  = hpt_scores$HPT_TOT9,
  HPT_POP   = hpt_scores$POP_raw,  # presentism foil (higher = worse)
  HPT_ROA   = hpt_scores$ROA,
  HPT_CONT  = hpt_scores$CONT
)

# optional blocks
kn_cols <- paste0("KN", 1:6); has_kn <- all(kn_cols %in% names(dat))
fr_cols <- c(paste0("RD", 1:3), paste0("NS", 1:3)); has_fr <- all(fr_cols %in% names(dat))
ksa_cols <- c(paste0("A",1:3), paste0("U",1:3), paste0("K",1:3)); has_ksa <- all(ksa_cols %in% names(dat))
sdr_cols <- paste0("SDR", 1:5); has_sdr <- all(sdr_cols %in% names(dat))

if (has_kn) {
```

```
  scales_list$KN_total <- rowSums(dat[keep, kn_cols], na.rm = TRUE)
}

if (has_fr) {
  fr_df <- dat[keep, fr_cols]
  scales_list$FR_RD     <- rowMeans(fr_df[, paste0("RD",1:3)], na.rm = TRUE)
  scales_list$FR_NS     <- rowMeans(fr_df[, paste0("NS",1:3)], na.rm = TRUE)
  scales_list$FR_total  <- rowMeans(fr_df, na.rm = TRUE)
}

if (has_ksa) {
  ksa_df <- dat[keep, ksa_cols]
  scales_list$KSA_A     <- rowMeans(ksa_df[, paste0("A",1:3)], na.rm = TRUE)
  scales_list$KSA_U     <- rowMeans(ksa_df[, paste0("U",1:3)], na.rm = TRUE)
  scales_list$KSA_K     <- rowMeans(ksa_df[, paste0("K",1:3)], na.rm = TRUE)
  scales_list$KSA_total <- rowMeans(ksa_df, na.rm = TRUE)
}

if (has_sdr) {
  sdr_df <- dat[keep, sdr_cols]
  scales_list$SDR_total <- rowMeans(sdr_df, na.rm = TRUE)
}

scales_df <- as_tibble(scales_list)

# Pairwise complete correlations
cors <- cor(scales_df, use = "pairwise.complete.obs")

print_tbl(round(cors, 3), caption = "Cross-construct correlations (pairwise complete).")
```

Table 5: Cross-construct correlations (pairwise complete).

	HPT_CTX6	HPT_TOT9	HPT_POP	HPT_ROA	HPT_CONT	KN_total	FR_RD	FR_NS	FR_total	KSA_A	KSA_U	KSA_K
HPT_CTX6	1.000	0.912	-0.802	0.322	0.844	0.321	-0.049	-0.006	-0.032	-0.016	-0.077	-0.029
HPT_TOT9	0.912	1.000	-0.685	0.683	0.811	0.370	-0.059	-0.021	-0.045	-0.005	-0.075	0.002
HPT_POP	-0.802	-0.685	1.000	-0.152	-0.357	-0.320	0.099	0.091	0.109	0.084	0.153	0.131
HPT_ROA	0.322	0.683	-0.152	1.000	0.368	0.282	-0.048	-0.037	-0.047	0.017	-0.035	0.056
HPT_CONT	0.844	0.811	-0.357	0.368	1.000	0.215	0.012	0.072	0.048	0.050	0.017	0.073
KN_total	0.321	0.370	-0.320	0.282	0.215	1.000	-0.062	-0.150	-0.129	0.014	0.023	0.087
FR_RD	-0.049	-0.059	0.099	-0.048	0.012	-0.062	1.000	0.438	0.841	0.395	0.411	0.358

FR_NS	-0.006	-0.021	0.091	-0.037	0.072	-0.150	0.438	1.000	0.855	0.350	0.322	0.207
FR_total	-0.032	-0.045	0.109	-0.047	0.048	-0.129	0.841	0.855	1.000	0.437	0.416	0.332
KSA_A	-0.016	-0.005	0.084	0.017	0.050	0.014	0.395	0.350	0.437	1.000	0.334	0.429
KSA_U	-0.077	-0.075	0.153	-0.035	0.017	0.023	0.411	0.322	0.416	0.334	1.000	0.365
KSA_K	-0.029	0.002	0.131	0.056	0.073	0.087	0.358	0.207	0.332	0.429	0.365	1.000
KSA_total	-0.051	-0.032	0.158	0.017	0.062	0.059	0.503	0.380	0.510	0.794	0.719	0.778
SDR_total	-0.046	-0.022	0.048	0.032	-0.029	0.001	-0.060	-0.268	-0.193	-0.182	-0.199	-0.032

12 Reproducibility appendix

```r
sessionInfo()
```

```
## R version 4.4.2 (2024-10-31)
## Platform: x86_64-pc-linux-gnu
## Running under: Ubuntu 24.04.3 LTS
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.12.0
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.12.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C               LC_TIME=cs_CZ.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=cs_CZ.UTF-8    LC_MESSAGES=en_US.UTF-8    LC_PAPER=cs_CZ.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C             LC_MEASUREMENT=cs_CZ.UTF-8 LC_IDENTIFICATION=C
##
## time zone: Europe/Prague
## tzcode source: system (glibc)
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] kableExtra_1.4.0   knitr_1.50         performance_0.15.1 lme4_1.1-38        Matrix_1.7-1       semTools_0.5-7
##  [7] lavaan_0.6-20      psych_2.4.12       lubridate_1.9.4    forcats_1.0.0      stringr_1.5.1      dplyr_1.1.4
## [13] purrr_1.1.0        readr_2.1.5        tidyr_1.3.1        tibble_3.2.1       ggplot2_4.0.1      tidyverse_2.0.0
##
## loaded via a namespace (and not attached):
##  [1] gtable_0.3.6         xfun_0.54            insight_1.4.2        lattice_0.22-5       tzdb_0.5.0
##  [6] quadprog_1.5-8       vctrs_0.6.5          tools_4.4.2          Rdpack_2.6.4         generics_0.1.3
```

```
## [11] stats4_4.4.2        parallel_4.4.2      sandwich_3.1-1      pkgconfig_2.0.3     lavaan.mi_0.1-0
## [16] RColorBrewer_1.1-3  S7_0.2.1            lifecycle_1.0.4     GPArotation_2024.3-1 compiler_4.4.2
## [21] farver_2.1.2        textshaping_0.4.1   mnormt_2.1.1        codetools_0.2-20    htmltools_0.5.8.1
## [26] yaml_2.3.10         pillar_1.10.0       nloptr_2.2.1        MASS_7.3-61         reformulas_0.4.1
## [31] boot_1.3-31         multcomp_1.4-28     nlme_3.1-166        tidyselect_1.2.1    digest_0.6.37
## [36] mvtnorm_1.3-2       stringi_1.8.4       labeling_0.4.3      splines_4.4.2       fastmap_1.2.0
## [41] grid_4.4.2          cli_3.6.5           magrittr_2.0.3      survival_3.7-0      pbivnorm_0.6.0
## [46] TH.data_1.1-4       withr_3.0.2         scales_1.4.0        estimability_1.5.1  timechange_0.3.0
## [51] rmarkdown_2.29      emmeans_1.10.6      zoo_1.8-14          hms_1.1.3           coda_0.19-4.1
## [56] evaluate_1.0.5      rbibutils_2.3       viridisLite_0.4.2   rlang_1.1.6         Rcpp_1.0.13-1
## [61] xtable_1.8-4        glue_1.8.0          xml2_1.3.6          svglite_2.2.2       rstudioapi_0.17.1
## [66] minqa_1.2.8         R6_2.6.1            systemfonts_1.3.1
```