Measurement checks — HPT (Czech data)

Reliability, dimensionality, presentism-contextualization contrast, and ICCs

HPT and Extremism project

2025-12-09

Contents

1 What this document does **2**

2 Setup and data loading **2**

3 Step 1 — Descriptives and scale construction **4**

4 Step 2 — Reliability: α and ω for POP-ROA-CONT **5**

5 Step 3 — Dimensionality (CFA/EFA) **6**

5.0.1 Optional: Data-driven EFA (polychoric) 9

6 Step 4 — Presentism-contextualization contrast (POP vs CONT) **11**

7 Step 5 — Distribution checks **12**

8 Step 6 — Class-level ICCs (is a multilevel model warranted?) **14**

9 Step 7 — Knowledge mini-test (KN1–KN6) **15**

10 Step 8 — Ideology batteries (FR-LF mini, KSA-3) and Social Desirability (SDR-5) **17**
10.1 FR-LF mini (RD1-RD3, NS1-NS3) . . . 17
10.2 KSA-3 (A1-A3, U1-U3, K1-K3) . . . 18
10.3 SDR-5 (SDR1-SDR5) . . . 20

11 Step 9 — Cross-construct correlations (HPT, KN, FR-LF, KSA-3, SDR-5) **21**

12 Reproducibility appendix **23**

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 the three HPT modes: **POP**, **ROA**, **CONT** (three items each; response scale 1-4).
2. **Dimensionality (CFA/EFA):** Does the **factor structure** match prior research (roughly, **POP+CONT together** versus **ROA** as a separate factor; or three distinct but correlated factors)?
3. **Presentism-contextualization contrast:** Do **POP** (presentist) and **CONT** (contextualization) show the expected contrast in the Czech data (differences in means / correlations)?
4. **Class-level clustering (ICCs):** Are scores clustered by **class** (so that multilevel models are justified later)?

Input: We assume a file `normalised_data.RData` providing an object `normalised_data` with variables `POP1-3`, `ROA1-3`, `CONT1-3`, and `class_label`.
Output: A human-readable PDF with tables/figures and short interpretations.

2 Setup and data loading

We load common R packages, then load the preprocessed dataset your pipeline already created.

```r
options(width = 120)

# Data handling & plots
library(tidyverse)

# Psychometrics
library(psych)        # alpha, omega, polychoric, EFA helpers
```

```r
library(lavaan)       # CFA
library(semTools)     # model comparisons & extras

# 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


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")
miss   <- setdiff(need, names(dat))
if (length(miss)) stop("Missing variables: ", paste(miss, collapse = ", "))

# Keep rows complete on HPT items for psychometric checks
hpt_items <- dat %>% select(all_of(hpt_cols)) %>% drop_na()
n_complete <- nrow(hpt_items)
cat("Rows with complete HPT data:", n_complete, "\n")
```

```
## Rows with complete HPT data: 148
```

We create an **analysis dataframe** keeping only rows with complete HPT data and a non-missing `class_label`.

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

analysis_df <- dat[keep, c(hpt_cols, "class_label")] %>%
  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): ", nrow_keep, "\n", sep = "")
```

```
## Rows in full data: 156
## Rows kept (complete HPT + class_label): 148
```

3 Step 1 — Descriptives and scale construction

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

- We compute subscale **means** for POP, ROA, CONT (each ranges 1-4).
- We also compute a grand **HPT_total** (mean of the three subscales).
- Then we print summaries and a quick correlation overview.

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

hpt_scores <- hpt_items %>%
  mutate(
    POP  = 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),
    HPT_total = rowMeans(across(c(POP, ROA, CONT)), na.rm = TRUE)
  )

summary(select(hpt_scores, POP, ROA, CONT, HPT_total))
```

```
##       POP             ROA             CONT          HPT_total
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:1.333   1st Qu.:2.333   1st Qu.:2.333   1st Qu.:2.333
##  Median :2.000   Median :3.000   Median :2.667   Median :2.556
##  Mean   :1.986   Mean   :2.813   Mean   :2.721   Mean   :2.507
```

```
##   3rd Qu.:2.417   3rd Qu.:3.333   3rd Qu.:3.333   3rd Qu.:2.778
##   Max.   :3.667   Max.   :4.000   Max.   :4.000   Max.   :3.333
```

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

```
##             POP        ROA       CONT
## POP   1.0000000 -0.1515091 -0.3757313
## ROA  -0.1515091  1.0000000  0.4127060
## CONT -0.3757313  0.4127060  1.0000000
```

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

Why: Reliability indicates whether items that are supposed to measure the same thing **hang together**. We report:

- α **(alpha)** on raw item data (common baseline), and
- α **and** ω from **polychoric** correlations (better for ordinal 1-4 items).

Interpretation tip for readers: $\omega_{\textbf{total}} \gtrsim .70$ is often seen as acceptable; $\omega_{\textbf{hier}}$ indicates strength of a general factor (useful if items might reflect a dominant common trait).

```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)
```

```r
  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
```

```r
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	0.510	0.565	0.582
ROA	3	0.530	0.557	0.598
CONT	3	0.644	0.702	0.709

How to read this table: Higher values mean items within a subscale are consistent. If a subscale shows **low** α **and** ω, consider revisiting items or treating the subscale cautiously in later analyses.

5 Step 3 — Dimensionality (CFA/EFA)

Goal: Check whether our data reproduce the **structure** reported in prior HPT work (often: **POP+CONT** vs **ROA**, or a **three-factor** model with POP, CONT, ROA as correlated factors).

We fit three **confirmatory factor models** using **ordered** items (WLSMV):

- **M1 (two factors):** *F1* loads on `POP1-3` and `CONT1-3`; *F2* loads on `ROA1-3`.
- **M2 (three factors):** *POP*, *CONT*, *ROA* as separate but correlated.
- **M3 (one factor):** Everything loads on a single general factor.

We then compare model fit and inspect loadings.

```r
hpt_ord <- hpt_items  # same data; we explicitly 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 side-by-side
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.
```

Now we print key indices and standardized loadings for each model.

```r
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)
)

# Print nicely
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
##          0.993          0.991          0.025          0.000          0.072          0.068
##
##
## |lhs |op |rhs   | est.std|    se|       z| pvalue| ci.lower| ci.upper|
## |:---|:--|:-----|-------:|-----:|-------:|------:|--------:|--------:|
## |F1  |=~ |POP1  |   0.588| 0.080|   7.307|  0.000|    0.430|    0.745|
## |F1  |=~ |POP2  |   0.243| 0.085|   2.842|  0.004|    0.075|    0.410|
## |F1  |=~ |POP3  |   0.416| 0.077|   5.374|  0.000|    0.264|    0.568|
## |F1  |=~ |CONT1 |  -0.720| 0.067| -10.684|  0.000|   -0.852|   -0.588|
## |F1  |=~ |CONT2 |  -0.606| 0.074|  -8.234|  0.000|   -0.750|   -0.462|
## |F1  |=~ |CONT3 |  -0.624| 0.072|  -8.715|  0.000|   -0.764|   -0.483|
## |F2  |=~ |ROA1  |   0.774| 0.100|   7.747|  0.000|    0.578|    0.969|
## |F2  |=~ |ROA2  |   0.302| 0.097|   3.102|  0.002|    0.111|    0.493|
## |F2  |=~ |ROA3  |   0.579| 0.089|   6.475|  0.000|    0.404|    0.755|
##
## ### 3-factor (POP/CONT/ROA)
##            cfi            tli          rmsea rmsea.ci.lower rmsea.ci.upper           srmr
##          1.000          1.037          0.000          0.000          0.024          0.051
##
##
## |lhs  |op |rhs   | est.std|    se|      z| pvalue| ci.lower| ci.upper|
## |:----|:--|:-----|-------:|-----:|------:|------:|--------:|--------:|
## |POP  |=~ |POP1  |   0.815| 0.111|  7.362|  0.000|    0.598|    1.032|
## |POP  |=~ |POP2  |   0.322| 0.092|  3.501|  0.000|    0.142|    0.502|
## |POP  |=~ |POP3  |   0.520| 0.080|  6.489|  0.000|    0.363|    0.677|
## |CONT |=~ |CONT1 |   0.742| 0.068| 10.871|  0.000|    0.609|    0.876|
```

```
## |CONT |=~ |CONT2 |   0.613| 0.074|  8.299|  0.000|    0.468|    0.757|
## |CONT |=~ |CONT3 |   0.642| 0.073|  8.847|  0.000|    0.500|    0.785|
## |ROA  |=~ |ROA1  |   0.772| 0.096|  8.061|  0.000|    0.584|    0.959|
## |ROA  |=~ |ROA2  |   0.308| 0.097|  3.176|  0.001|    0.118|    0.498|
## |ROA  |=~ |ROA3  |   0.578| 0.088|  6.567|  0.000|    0.406|    0.751|
##
## ### 1-factor (general)
##            cfi            tli          rmsea rmsea.ci.lower rmsea.ci.upper           srmr
##          0.962          0.949          0.059          0.012          0.094          0.080
##
##
## |lhs |op |rhs   | est.std|    se|       z| pvalue| ci.lower| ci.upper|
## |:---|:--|:-----|-------:|-----:|-------:|------:|--------:|--------:|
## |G   |=~ |POP1  |   0.571| 0.080|   7.137|  0.000|    0.414|    0.728|
## |G   |=~ |POP2  |   0.224| 0.084|   2.652|  0.008|    0.058|    0.390|
## |G   |=~ |POP3  |   0.401| 0.078|   5.169|  0.000|    0.249|    0.554|
## |G   |=~ |ROA1  |  -0.575| 0.082|  -7.061|  0.000|   -0.735|   -0.416|
## |G   |=~ |ROA2  |  -0.230| 0.087|  -2.634|  0.008|   -0.402|   -0.059|
## |G   |=~ |ROA3  |  -0.467| 0.077|  -6.082|  0.000|   -0.617|   -0.316|
## |G   |=~ |CONT1 |  -0.698| 0.064| -10.969|  0.000|   -0.823|   -0.574|
## |G   |=~ |CONT2 |  -0.595| 0.072|  -8.298|  0.000|   -0.735|   -0.454|
## |G   |=~ |CONT3 |  -0.606| 0.068|  -8.846|  0.000|   -0.740|   -0.471|
```

How to interpret: Prefer models with **CFI/TLI** $\gtrsim$ .95, **RMSEA** $\lesssim$ .06-.08, **SRMR** $\lesssim$ .08 (rules of thumb). If the 2- or 3-factor model clearly outperforms 1-factor and loadings align with expectations (POP & CONT together; ROA separate—or all three distinct), the data support the theorized structure.

5.0.1 Optional: Data-driven EFA (polychoric)

Why: As a sensitivity check, we can inspect **exploratory** factor analysis using polychoric correlations.

```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  -0.347  0.420
## POP2          0.563
## POP3          0.489
## ROA1   0.700
## ROA2   0.425  0.289
## ROA3   0.552
## CONT1  0.613
## CONT2  0.460 -0.270
## CONT3  0.499
##
##                  PA1   PA2
## SS loadings    1.954 0.978
## Proportion Var 0.217 0.109
## Cumulative Var 0.217 0.326
```

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

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

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

```
##
## Loadings:
##       PA1    PA2    PA3
## POP1          0.519
## POP2          0.431  0.289
## POP3          0.676
## ROA1                 0.611
## ROA2                 0.472
## ROA3                 0.516
## CONT1  0.725
## CONT2  0.414
## CONT3  0.716
##
```

```
##                    PA1   PA2   PA3
## SS loadings    1.276 0.980 0.985
## Proportion Var 0.142 0.109 0.109
## Cumulative Var 0.142 0.251 0.360
```

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

Idea: Prior work suggests **presentist** choices (POP) and **contextualized** reasoning (CONT) should **pull in opposite directions**. Here we check whether the **Czech data** replicate that **contrast**: (a) compare means; (b) inspect the POP-CONT correlation.

```
# Rebuild subscale scores locally to avoid scope/version issues
hpt_scores_local <- hpt_items %>%
  mutate(
    POP  = 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),
    HPT_total = rowMeans(across(c(POP, ROA, CONT)), na.rm = TRUE)
  )

# Sanity check: make sure the columns exist
stopifnot(all(c("POP","ROA","CONT","HPT_total") %in% names(hpt_scores_local)))

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

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

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

mean_POP	sd_POP	mean_CONT	sd_CONT	r_POP_CONT
1.986	0.67	2.721	0.739	-0.376

```
# Simple paired comparison (descriptive; not a preregistered test)
t_test <- t.test(hpt_scores_local$POP, hpt_scores_local$CONT, paired = TRUE)
t_test
```

```
##
##  Paired t-test
##
## data:  hpt_scores_local$POP and hpt_scores_local$CONT
## t = -7.6411, df = 147, p-value = 2.555e-12
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  -0.9241307 -0.5443378
## sample estimates:
## mean difference
##      -0.7342342
```

Reading the results:

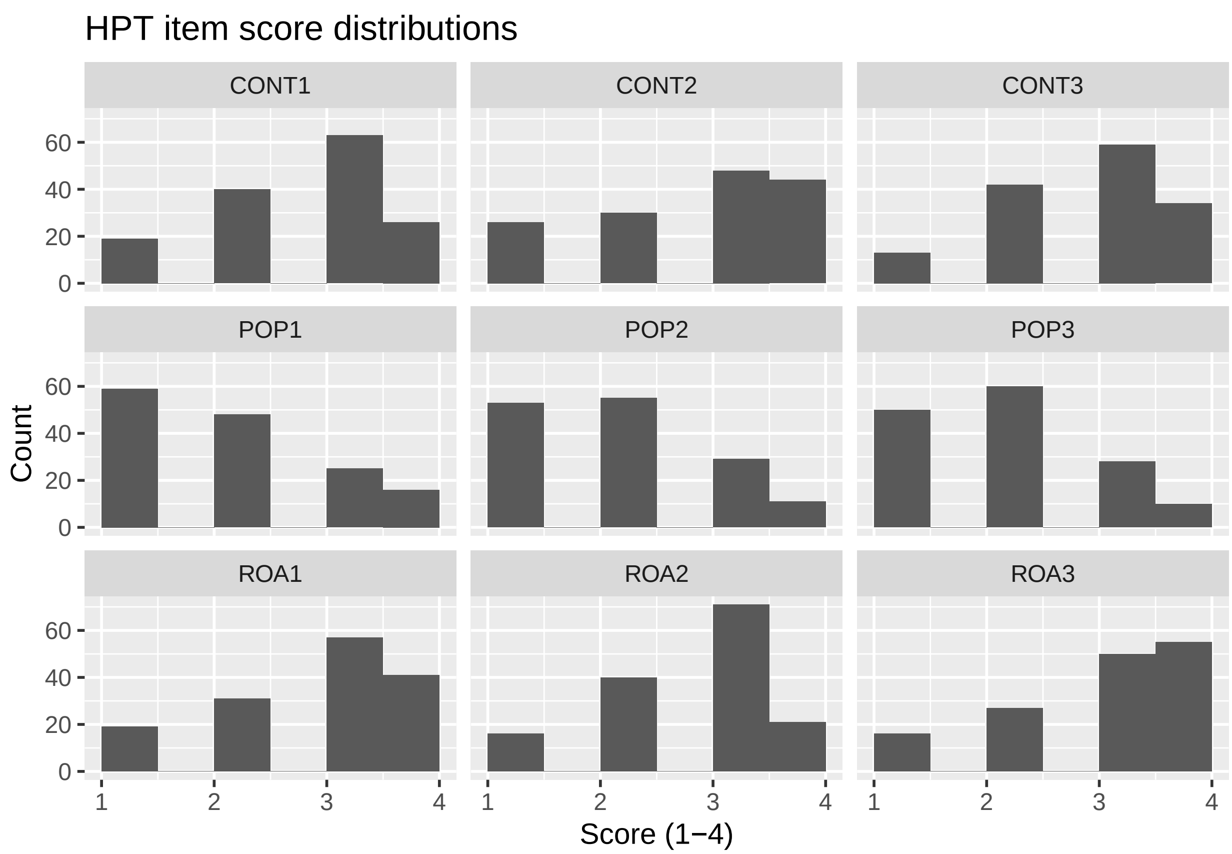
- If **mean_CONT** > **mean_POP** and/or $r_{\mathbf{POP,CONT}} < 0$, that supports the expected contrast.
- If they move **together** (positive correlation and similar means), interpretation of the HPT construct may require caution.

7 Step 5 — Distribution checks

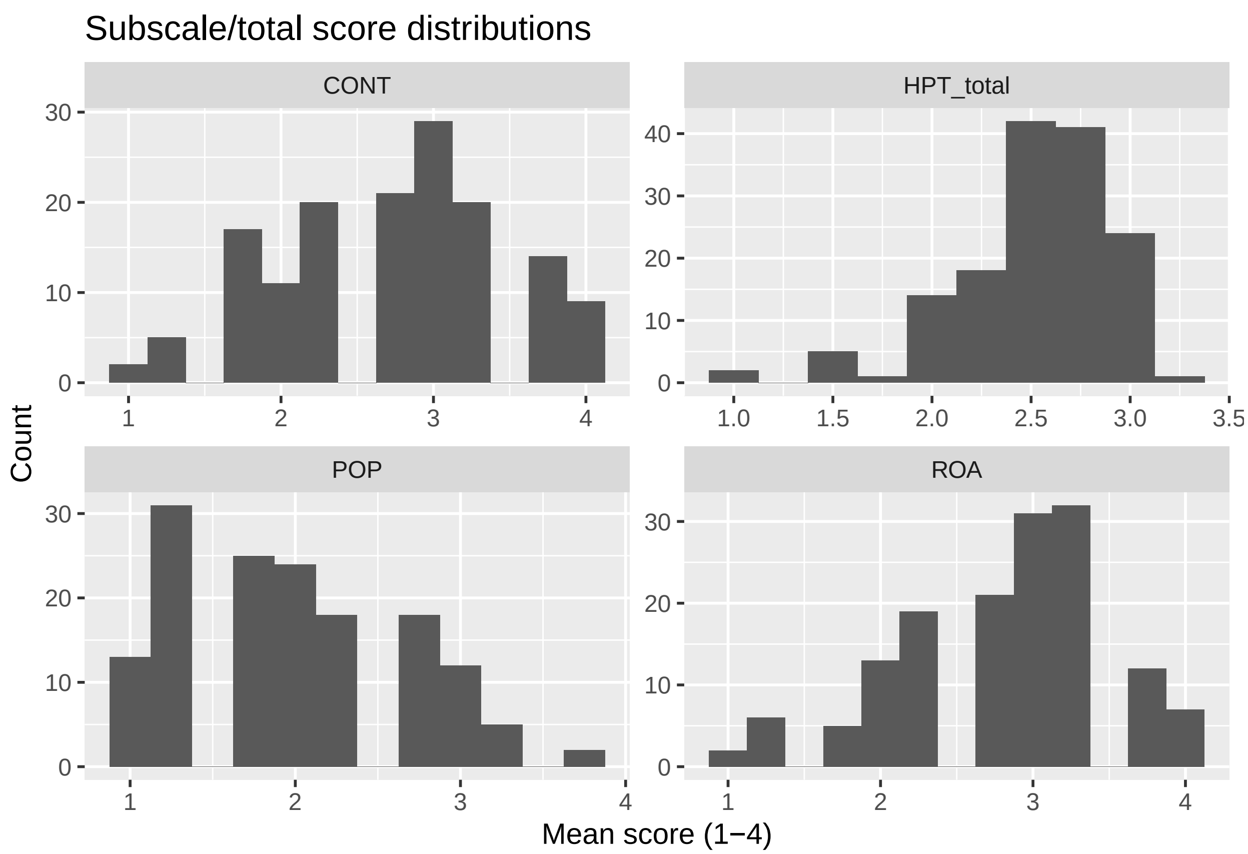
Why: Skewed or piled-up scores can cause model or inference issues. We look at item-level and scale-level histograms.

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

# Item distributions
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 distributions
ggplot(hpt_scores %>% pivot_longer(c(POP, ROA, CONT, HPT_total), names_to = "scale", values_to = "score"),
       aes(x = score)) +
  geom_histogram(binwidth = 0.25) +
  facet_wrap(~ scale, scales = "free") +
  labs(title = "Subscale/total score distributions", x = "Mean score (1-4)", y = "Count")
```



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

Why: Students are nested in **classes**; scores may be more similar within a class. The **Intraclass Correlation Coefficient (ICC)** estimates the fraction of variance at the class level. If ICC $\gtrsim .05$, multilevel modeling is usually advisable.

We fit **null (random-intercept)** models for **HPT_total**, **POP**, **ROA**, **CONT** and extract ICCs.

```
# analysis_df and hpt_scores already exist and are aligned
icc_data <- analysis_df %>%
  transmute(class_label) %>%
  bind_cols(hpt_scores %>% select(POP, ROA, CONT, HPT_total))

mk_icc <- function(dv){
```

```r
  f <- reformulate("1 + (1|class_label)", response = dv)
  fit <- lmer(f, data = icc_data, REML = TRUE)

  # Extract variance components
  vc <- as.data.frame(VarCorr(fit))
  var_class <- vc$vcov[vc$grp == "class_label"][1]
  var_resid <- vc$vcov[vc$grp == "Residual"][1]

  icc <- var_class / (var_class + var_resid)

  tibble(
    DV = dv,
    ICC = icc,
    var_class = var_class,
    var_resid = var_resid,
    singular = isSingular(fit)
  )
}

icc_tbl <- purrr::map_dfr(c("HPT_total","POP","ROA","CONT"), mk_icc)

print_tbl(icc_tbl, digits = 3, caption = "Null-model ICCs by outcome (computed from variance components).")
```

Table 3: Null-model ICCs by outcome (computed from variance components).

DV	ICC	var_class	var_resid	singular
HPT_total	0.016	0.002	0.149	FALSE
POP	0.013	0.006	0.444	FALSE
ROA	0.011	0.005	0.458	FALSE
CONT	0.055	0.030	0.517	FALSE

Interpretation:

- **Higher ICC** ⇒ more clustering by class.
- Non-trivial ICCs motivate **multilevel** models for confirmatory analyses.

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

Why: The KN items are dichotomous (0/1). We report:

- KR-20 (equivalent to alpha for dichotomous items)
- Item difficulty (p = proportion correct)
- Point-biserial discrimination (w.r.t. total score)

```r
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	3.297	1.639	0.571

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

Why:

- We need reliable predictors and controls before hypothesis tests.
- We report α/ω (polychoric), optional CFA fits, and descriptive summaries.

```
# 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)
    },
    {
      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)
```

```r
    },
    {
      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
## 0.998 0.996 0.020 0.051
```

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))

if (!has_ksa) {
  cat("\n**KSA-3 section skipped:** A1-A3, U1-U3, and/or K1-K3 not found.\n")
} else {
```

```r
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)
if (!inherits(ksa_fit, "try-error")) {
  print(fitMeasures(ksa_fit, c("cfi","tli","rmsea","srmr")))
} else {
```

```r
    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
## 0.978 0.968 0.062 0.071
```

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()
  # NOTE: Your data reportedly already has SDR2-SDR4 reversed. If unsure, you can
  # check symmetry and optionally reverse here before reliability.
  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
## 0.739 0.478 0.210 0.114
```

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

Why: Useful overview to see how constructs relate before multilevel models.

```
# Build scale scores that exist in your data (gracefully skipping any missing block)
scales_list <- list(
  HPT_total = hpt_scores$HPT_total,
  HPT_POP   = hpt_scores$POP,
  HPT_ROA   = hpt_scores$ROA,
  HPT_CONT  = hpt_scores$CONT
)

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)
```

```r
  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_total	HPT_POP	HPT_ROA	HPT_CONT	KN_total	FR_RD	FR_NS	FR_total	KSA_A	KSA_U	KSA_K	KSA_total
HPT_total	1.000	0.248	0.757	0.658	0.128	-0.041	0.015	-0.017	0.111	0.091	0.192	0.169
HPT_POP	0.248	1.000	-0.152	-0.376	-0.306	0.082	0.098	0.109	0.081	0.069	0.153	0.127
HPT_ROA	0.757	-0.152	1.000	0.413	0.262	-0.059	-0.054	-0.066	0.041	0.046	0.093	0.079
HPT_CONT	0.658	-0.376	0.413	1.000	0.239	-0.085	-0.015	-0.065	0.064	0.037	0.079	0.079
KN_total	0.128	-0.306	0.262	0.239	1.000	-0.131	-0.203	-0.198	-0.025	-0.007	0.017	-0.001
FR_RD	-0.041	0.082	-0.059	-0.085	-0.131	1.000	0.405	0.842	0.402	0.414	0.366	0.502
FR_NS	0.015	0.098	-0.054	-0.015	-0.203	0.405	1.000	0.833	0.388	0.305	0.204	0.385
FR_total	-0.017	0.109	-0.066	-0.065	-0.198	0.842	0.833	1.000	0.470	0.429	0.341	0.528
KSA_A	0.111	0.081	0.041	0.064	-0.025	0.402	0.388	0.470	1.000	0.365	0.439	0.800
KSA_U	0.091	0.069	0.046	0.037	-0.007	0.414	0.305	0.429	0.365	1.000	0.419	0.736
KSA_K	0.192	0.153	0.093	0.079	0.017	0.366	0.204	0.341	0.439	0.419	1.000	0.792
KSA_total	0.169	0.127	0.079	0.079	-0.001	0.502	0.385	0.528	0.800	0.736	0.792	1.000
SDR_total	0.100	0.107	0.005	0.057	0.014	-0.033	-0.197	-0.133	-0.148	-0.177	-0.012	-0.144

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=C.UTF-8           LC_NUMERIC=C               LC_TIME=cs_CZ.UTF-8        LC_COLLATE=C.UTF-8
##  [5] LC_MONETARY=cs_CZ.UTF-8    LC_MESSAGES=C.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
```