```
################################################################################
#  SMARTstats Workshop: Confirmatory Factor Analysis (CFA)
#  Foundations of CFA with lavaan in R
#  Companion script to the SMARTstats CFA slide deck
#
#  Structure:
#    Part 0: Setup & Data Prep                          (~3  min)
#    Part 1: EFA Recap on Full Sample                   (~3  min)
#    Part 2: Split-Half & Specifying the CFA Model      (~3  min)
#    Part 3: Fitting and Reading Output                 (~7  min)
#    Part 4: Local Fit & the Respecification Decision   (~12 min)  *** DECISION
POINT ***
#    Part 5: Validating on the Holdout Sample           (~5  min)
#    Part 6: Cameo - When Default ML Isn't Enough       (~2  min)
#    Part 7: Reporting Checklist                        (~1  min)
################################################################################


# ==============================================================================
# PART 0: SETUP & DATA PREP  (~3 min)
# ==============================================================================

# --- Install packages (run once) ---
# install.packages(c("lavaan", "semTools", "psych", "dplyr"))

# --- Load packages ---
library(lavaan)     # workhorse for CFA / SEM
library(semTools)   # auxiliary tools (reliability, invariance, etc.)
library(psych)      # descriptives, EFA, alpha, etc.
library(dplyr)      # data wrangling


# --- Read data ---
# NOTE: Update this path. The data file is the IPIP Big Five (50 items)
# distributed at https://openpsychometrics.org/_rawdata/BIG5.zip and
# mirrored as "big5.csv" in the workshop materials.
# setwd("path/to/workshop/materials")
dat <- read.csv("big5.csv")

# Quick look
dim(dat)         # ~19,719 rows
str(dat[, 1:10])
head(dat[, 1:10])

# The first 7 columns are demographics; columns 8-57 are the 50 IPIP items
# (E1-E10, N1-N10, A1-A10, C1-C10, O1-O10), each rated 1-5.
# A response of 0 indicates a missing item; recode to NA before analysis.

items <- dat[, 8:57]
items[items == 0] <- NA
```

```r
# --- Reverse-code negatively-keyed items ---
# In the IPIP-50, several items are phrased so that a HIGH response reflects
# a LOW level of the trait (e.g., "I don't talk a lot" -> high score = low E).
# We reverse-code so that, for every item, a HIGH score = HIGH level of the
# trait. This makes both the EFA loadings and the CFA solution easier to
# read (all loadings expected positive).
#
# Standard IPIP-50 reverse-keyed items:
#   Extraversion (E):       E2, E4, E6, E8, E10
#   Neuroticism (N):        N2, N4
#   Agreeableness (A):      A1, A3, A5, A7
#   Conscientiousness (C):  C2, C4, C6, C8
#   Openness (O):           O2, O4, O6

reverse_items <- c("E2","E4","E6","E8","E10",
                   "N2","N4",
                   "A1","A3","A5","A7",
                   "C2","C4","C6","C8",
                   "O2","O4","O6")

# Likert items run 1-5; reverse means new = 6 - old
items[, reverse_items] <- 6 - items[, reverse_items]

# Sanity check 1: a positively-keyed item (E1) and a reverse-keyed item (E2)
# should now correlate POSITIVELY after recoding.
cor(items$E1, items$E2, use = "pairwise.complete.obs")

# Sanity check 2: two positively-keyed items (E1, E3) for comparison.
# Should be in roughly the same range as the recoded pair above.
cor(items$E1, items$E3, use = "pairwise.complete.obs")
# If both are positive and similar in magnitude, recoding put all items
# on the same scale.


# --- Note on missing data ---
# We use FIML (Full Information Maximum Likelihood) in all cfa() calls
# below. Every case contributes to estimation for the variables it has,
# even with some items missing. No listwise deletion needed.
#
# Quick look at the missingness pattern:
sum(!complete.cases(items))                 # number of incomplete rows
mean(complete.cases(items))                 # proportion fully complete
colSums(is.na(items)) |> sort(decreasing = TRUE) |> head(10)  # most-missing
dim(items)



# ============================================================================
# PROLOGUE: EFA RECAP ON FULL SAMPLE  (~3 min)
# ============================================================================
```

```
# Brief recap to motivate the CFA model. The prior workshop covered EFA
# methodology in depth.

#The EFA shown here is illustrative only.
#It is not used to discover the CFA model, because the Big Five model is theory-
specified in advance.

# --- Eigenvalues to confirm 5 factors ---
# Look at the first ~10 eigenvalues of the correlation matrix.
# A clean Big Five solution shows 5 eigenvalues clearly above the noise
# floor, then a sharp drop.
ev <- eigen(cor(items, use = "pairwise.complete.obs"))$values
round(ev[1:10], 2)
# Expected pattern: roughly 7.4 / 4.1 / 3.1 / 3.0 / 2.4 / then drops below 1.
# Expected pattern not observed, eigenvalues 6-8 exceed 1.
# Optional: scree plot
plot(ev[1:15], type = "b", pch = 19,
     xlab = "Factor", ylab = "Eigenvalue",
     main = "Scree plot (first 15 factors)")
abline(h = 1, lty = 2, col = "red")


# --- 5-factor EFA with oblique (oblimin) rotation ---
# Oblique because Big Five factors are theoretically allowed to correlate.
# use = "pairwise" handles NAs without listwise deletion.
efa_fit <- fa(items, nfactors = 5, rotate = "oblimin", fm = "ml",
              use = "pairwise")

# Print loadings, suppressing values below .30 and sorting by factor.
# Each item should load primarily on its expected factor.
print(efa_fit$loadings, cutoff = 0.30, sort = TRUE)


# --- What we learn from the EFA ---
# The first 5 factors correspond to the canonical Big Five
# (Extraversion, Neuroticism, Agreeableness, Conscientiousness, Openness).
# Items group mostly as the theory predicts, with some exceptions (low loadings
for N4). This justifies specifying the same 5-factor structure as our hypothesized
CFA model in Part 2.
#
# Reminder: EFA is exploratory by design; the same dataset cannot then
# *confirm* the structure it suggested. So in Part 2 we split the sample
# and run the CFA on a derivation half, validating on a holdout half.



# ============================================================================
# PART 2: SPLIT-HALF & SPECIFYING THE CFA MODEL  (~3 min)
# ============================================================================
```

```
# --- Split-half: derivation vs. holdout ---
# We CONFIRM the 5-factor structure suggested by EFA. To avoid circularity
# and to test whether any respecification decisions generalize, we split
# the data into two random halves: derivation (where we fit and may
# respecify) and holdout (untouched until Part 5).

set.seed(8675309)                                  # any fixed seed; document it
n     <- nrow(items)
idx   <- sample(seq_len(n), size = floor(n/2))   # ~50% for derivation

deriv  <- items[idx, ]      # derivation half: where we fit and modify
holdout <- items[-idx, ]    # holdout half: untouched until Part 5

dim(deriv)
dim(holdout)



# --- Model from theory + the EFA recap above ---
# Big Five theory specifies 5 correlated factors with 10 indicators each.

#
# Notes on what we are NOT doing in the initial model:
#   - No cross-loadings (each item loads on exactly one factor)
#   - No correlated residuals
#   - Factors are allowed to correlate freely
#   - Default identification: marker method (first loading fixed to 1)


# --- lavaan model syntax cheat sheet ---
#   =~   "is measured by"       e.g.   E =~ E1 + E2 + E3
#   ~~   covariance              e.g.  E1 ~~ E2   (correlated residuals)
#   ~    regression               e.g.   y ~ x
#   ~1   intercept/mean
#   1*   fix loading/parameter to 1
#   NA*  free a fixed parameter
#   a*   label parameter "a" for constraints


# --- Specify the 5-factor measurement model ---
big5_model <- '
  # Latent factors and their indicators
  E =~ E1 + E2 + E3 + E4 + E5 + E6 + E7 + E8 + E9 + E10
  N =~ N1 + N2 + N3 + N4 + N5 + N6 + N7 + N8 + N9 + N10
  A =~ A1 + A2 + A3 + A4 + A5 + A6 + A7 + A8 + A9 + A10
  C =~ C1 + C2 + C3 + C4 + C5 + C6 + C7 + C8 + C9 + C10
  O =~ O1 + O2 + O3 + O4 + O5 + O6 + O7 + O8 + O9 + O10
'
# Factor covariances and residual variances are added automatically by cfa().
```

```
# ============================================================================
# PART 3: FITTING AND READING OUTPUT  (~7 min)
# ============================================================================

# --- Fit the model on the DERIVATION sample ---
# We use FIML (missing = "fiml") so every case contributes to estimation,
# even with some items missing. With FIML, lavaan also estimates the
# 50 indicator means/intercepts -- this enlarges the parameter count
# and the df accounting (see the output header) but is the correct
# treatment when the data have any missingness.
fit1 <- cfa(big5_model,
            data       = deriv,
            estimator  = "ML",
            missing    = "fiml",
            std.lv     = FALSE)     # marker method (first loading = 1)

# --- Read the output ---
summary(fit1, fit.measures = TRUE, standardized = TRUE)


# Walk-through: what to look at, and in what order.
#
# 1) HEADER
#    - "Number of free parameters" and "Number of observations" -- sanity check.
#    - With FIML on, free parameters include the 50 indicator intercepts in
#      addition to loadings, factor variances/covariances, and residuals.
#    - With 50 items, p(p+1)/2 = 1275 unique covariances/variances; FIML
#      also uses the 50 means -- 1325 known values total.
#    - The exact df depends on the missingness pattern but should be roughly
#      in the 1100-1200 range for this model.
#
# 2) MODEL TEST USER MODEL (model chi-square)
#    - With n ~ 9,800 the chi-square will be large and the p-value tiny.
#      DO NOT reject on chi-square alone.
#    - Chi-square is informative for nested-model COMPARISON, not absolute fit.
#
# 3) APPROXIMATE FIT INDICES
#    - CFI / TLI: closer to 1 is better. Hu & Bentler (1999) suggested >= .95;
#      contemporary work (McNeish & Wolf 2023) shows fixed cutoffs misbehave.
#    - RMSEA with 90% CI and p-close: smaller is better.
#    - SRMR: average standardized residual; smaller is better.
#    - Report all four. Do not cherry-pick.
#
# 4) PARAMETER ESTIMATES
#    - Look at "Std.all" loadings (fully standardized).
#      In a healthy CFA we expect loadings >= ~.40 for behavioral items.
#    - Inspect factor correlations under "Covariances".
#      If two factors correlate > .85, you may not have two distinct factors.
#    - Inspect residual variances. Negative residuals = Heywood case = trouble.
```

```
# Quick extraction helpers:
fitMeasures(fit1, c("chisq","df","pvalue","cfi","tli",
                    "rmsea","rmsea.ci.lower","rmsea.ci.upper",
                    "srmr"))

# Standardized loadings only:
inspect(fit1, "std")$lambda

# Factor correlations only:
lavInspect(fit1, "cor.lv")



# ============================================================================
# PART 4: LOCAL FIT AND THE RESPECIFICATION DECISION  (~12 min)
# *** DECISION POINT ***
# ============================================================================

# Global fit indices are an average. They can hide localized misfit. Two tools
# diagnose WHERE the model fits poorly:
#   (a) standardized residual covariances
#   (b) modification indices (MIs)


# --- (a) Standardized residuals ---
# Values |z| > ~2.5 indicate item pairs the model is failing to reproduce well.
res <- residuals(fit1, type = "standardized")
res_mat <- res$cov
diag(res_mat) <- 0
# Top 10 largest absolute standardized residuals:
top_resid <- as.data.frame(as.table(abs(res_mat)))
top_resid <- top_resid[order(-top_resid$Freq), ]
head(top_resid, 10)


# --- (b) Modification indices ---
# An MI is the EXPECTED chi-square reduction if a fixed parameter were freed.
# A large MI flags a parameter the data "want" to estimate.
mi <- modindices(fit1, sort = TRUE, maximum.number = 15)
print(mi)

# Two patterns commonly appear with the IPIP Big Five:
#
#   (i)  MIs suggesting CROSS-LOADINGS (e.g., E item loading on N).
#        Some are theoretically defensible; most are not.
#
#   (ii) MIs suggesting CORRELATED RESIDUALS between items that share
#        wording structure -- often a method effect of REVERSE keying
#        or near-duplicate item content. Well-documented in the Big Five
#        psychometric literature.
```

```
# *** THE DECISION POINT ***
#
# When an MI is large, the choice is NOT automatic. Three criteria for a
# defensible respecification:
#   1. Is there a SUBSTANTIVE rationale (theory, item content, known method
#      effect) -- independent of the MI itself?
#   2. Would you have predicted this parameter BEFORE seeing the MI?
#   3. Will you report and justify it transparently in the paper?
#
# If any answer is "no", do NOT free the parameter. MI-driven fishing for
# fit is the most common form of capitalization on chance in CFA.
#
# For demonstration we free ONE error covariance with a clear substantive
# rationale: the highest-MI residual covariance between two items whose
# content is near-duplicate. (Check your top MIs and pick a pair whose
# wording overlap or shared method is obvious. The exact pair will vary
# slightly by random split.)
#
# Below we use the MI output above to pick a defensible respecification.
# Replace with TWO sets of items your output identifies as a content-
# overlap pair.

big5_model_v2 <- '
  E =~ E1 + E2 + E3 + E4 + E5 + E6 + E7 + E8 + E9 + E10
  N =~ N1 + N2 + N3 + N4 + N5 + N6 + N7 + N8 + N9 + N10
  A =~ A1 + A2 + A3 + A4 + A5 + A6 + A7 + A8 + A9 + A10
  C =~ C1 + C2 + C3 + C4 + C5 + C6 + C7 + C8 + C9 + C10
  O =~ O1 + O2 + O3 + O4 + O5 + O6 + O7 + O8 + O9 + O10

  # Method-effect residual covariance (replace with your top 2 MI pairs)
  N7 ~~ N8
  O1 ~~ O8

'


fit2 <- cfa(big5_model_v2,
            data      = deriv,
            estimator = "ML",
            missing   = "fiml")
summary(fit2, fit.measures = TRUE, standardized = TRUE)


# --- Compare nested models ---
# fit1 is nested within fit2 (fit2 has two more free parameters).
# anova() gives the chi-square difference test.
anova(fit1, fit2)

# Compare global fit indices side by side:
round(rbind(
```

```
  fit1 = fitMeasures(fit1, c("chisq","df","cfi","tli","rmsea","srmr")),
  fit2 = fitMeasures(fit2, c("chisq","df","cfi","tli","rmsea","srmr"))
), 3)

# QUESTION FOR THE ROOM:
# The chi-square test almost certainly says fit2 is "better." But did we gain
# anything that matters? Did CFI go from .87 to .95? Or from .92 to .921?
# That is the substantive question, not the chi-square question.
#
# Lesson: every freed parameter is a story you owe the reader. If you cannot
# tell that story without pointing at the MI, do not free the parameter.
# In this case, N7-8 (mood swing) , O1-O8 (Difficult word enjoyers) are near
paraphrases of constructs not explained by the latent factor



# ============================================================================
# PART 5: VALIDATING ON THE HOLDOUT SAMPLE  (~5 min)
# ============================================================================

# Now we fit our final model on the holdout half. Two questions:
#   1. Does fit hold up on a sample we did NOT use for respecification?
#   2. Are the parameter estimates STABLE across the two halves?

fit_holdout <- cfa(big5_model_v2,
                   data      = holdout,
                   estimator = "ML",
                   missing   = "fiml")
summary(fit_holdout, fit.measures = TRUE, standardized = TRUE)

# Side-by-side fit comparison: derivation vs. holdout
round(rbind(
  derivation = fitMeasures(fit2,
c("chisq","df","cfi","tli","rmsea","srmr")),
  holdout    = fitMeasures(fit_holdout,
c("chisq","df","cfi","tli","rmsea","srmr"))
), 3)



# ============================================================================
# PART 6: CAMEO - WHEN DEFAULT ML ISN'T ENOUGH  (~2 min)
# ============================================================================

# Default ML assumes:
#   1. Continuous indicators
#   2. Multivariate normality
#   3. Large N (asymptotic)
#
# When (1) holds but (2) does not, use ROBUST ML:
fit2_mlr <- cfa(big5_model_v2,
```

```
                data      = deriv,
                estimator = "MLR",
                missing   = "fiml")
fitMeasures(fit2_mlr, c("chisq.scaled","df","cfi.robust",
                        "tli.robust","rmsea.robust","srmr"))
# MLR returns Satorra-Bentler-style adjustments to chi-square and SEs
# that are robust to non-normality.

# When indicators are ORDINAL (e.g., Likert with few categories) and you
# want to take that seriously, use WLSMV:
fit2_wlsmv <- cfa(big5_model_v2, data = deriv, estimator = "WLSMV",
                  ordered = names(deriv))
fitMeasures(fit2_wlsmv, c("chisq.scaled","df","cfi.robust",
                        "tli.robust","rmsea.robust","srmr"))
# WLSMV treats items as ordered categorical, fits a polychoric correlation
# model, and is the modern standard for ordinal CFA. It is slower and
# requires more sample size per item-pair, but it is correct in a way that
# treating Likert as continuous is not.
# Note: WLSMV uses pairwise deletion by default, not FIML, because it
# operates on polychoric correlations rather than the raw likelihood.
#
# Bottom line: report the estimator you used and why. Reviewers ask.



# ============================================================================
# PART 7: REPORTING CHECKLIST  (~1 min)
# ============================================================================

# A defensible CFA writeup contains, at minimum:
#
# [ ] N, source of data, missingness handling
# [ ] Estimator used and rationale (ML / MLR / WLSMV)
# [ ] Identification strategy (marker vs. variance standardization)
# [ ] Hypothesized model (figure or full lavaan syntax in supplement)
# [ ] Global fit: chi-square (df, p), CFI, TLI, RMSEA with 90% CI, SRMR
# [ ] Standardized factor loadings for all items
# [ ] Factor variances and factor correlations
# [ ] Any respecifications, with substantive justification (NOT "MI was big")
# [ ] If you split-validated, report fit and parameter stability on holdout
# [ ] Code and data (or seed + DGP) sufficient for replication
#
# Common issues reviewers will flag:
#   - Reporting CFI / RMSEA without SRMR
#   - Reporting fit indices without 90% CI on RMSEA
#   - Respecifying based on MIs without substantive justification
#   - Treating ordinal items as continuous without comment
#   - Selecting fit-index cutoffs without engaging with current debate

# End of script.
################################################################################
```