L&T PxV Measurement invariance

John Flournoy

10 April, 2018

Contents

Creating factor-item mapping

As part of this process, outputting the alpha coefficients.

```
## **Wave 1**
```

```
##          aspfin    bfa_mt bfas_ac   bfas_ap   bfas_ci   bfas_co   bfas_ea
## alpha 0.8007216 0.7954608 0.88408 0.7814323 0.8559713 0.7748649 0.8882828
##         bfas_ee   bfas_nv  bfas_nv9   bfas_nw   bfas_oi   bfas_oo
## alpha 0.8547331 0.9113937 0.9010348 0.8781388 0.8287349 0.7899743
##           bfi_a    bfi_a6     bfi_c bfi_d_scale     bfi_e   bfi_hp8
## alpha 0.8009146 0.7908462 0.819993   0.7966031 0.8749101 0.6762593
##           bfi_n     bfi_o bfi_s_scale   hrz_col  hrz_ind       usi
## alpha 0.8582059 0.7898225   0.7969331 0.7126198 0.550577 0.6732525
##         vrt_col   vrt_ind
## alpha 0.6743606 0.6523363
```

```
## **Wave 2**
```

```
##          aspfin    bfa_mt   bfas_ac   bfas_ap   bfas_ci   bfas_co
## alpha 0.8252445 0.7878581 0.8829338 0.7785236 0.8492025 0.7839002
##         bfas_ea   bfas_ee   bfas_nv  bfas_nv9   bfas_nw   bfas_oi
## alpha 0.8761572 0.8590842 0.8972032 0.8854566 0.8698163 0.8153715
##         bfas_oo     bfi_a    bfi_a6     bfi_c bfi_d_scale     bfi_e
## alpha 0.7970451 0.7962796 0.7961057 0.8071378   0.7858279 0.8664445
##         bfi_hp8     bfi_n     bfi_o bfi_s_scale   hrz_col   hrz_ind
## alpha 0.6848438 0.8587661 0.7966047   0.7848152 0.6932522 0.5938064
##             usi   vrt_col   vrt_ind
## alpha 0.6929115 0.7115297 0.6393136
```

```
## **Wave 3**
```

```
##          aspfin    bfa_mt   bfas_ac   bfas_ap   bfas_ci   bfas_co
## alpha 0.8338066 0.8046391 0.8862414 0.7768773 0.8492295 0.7774236
##         bfas_ea   bfas_ee   bfas_nv  bfas_nv9   bfas_nw   bfas_oi
## alpha 0.8819575 0.8616749 0.9007862 0.8903142 0.8817477 0.8376555
##         bfas_oo     bfi_a    bfi_a6     bfi_c bfi_d_scale     bfi_e
```

```
## alpha 0.8028977 0.803164 0.7982502 0.8213291   0.7909743 0.8728148
##         bfi_hp8     bfi_n     bfi_o bfi_s_scale   hrz_col   hrz_ind
## alpha 0.6738761 0.8582405 0.8133423   0.7918976 0.7079314 0.5391312
##             usi   vrt_col   vrt_ind
## alpha 0.6881401 0.6990938 0.6752587
```

```
## **Wave 4**
```

```
##          aspfin    bfa_mt   bfas_ac   bfas_ap   bfas_ci   bfas_co  bfas_ea
## alpha 0.8347909 0.8156071 0.8874039 0.7633623 0.8518579 0.7937478 0.881371
##         bfas_ee   bfas_nv  bfas_nv9   bfas_nw   bfas_oi   bfas_oo
## alpha 0.8705236 0.9131165 0.9049279 0.8847419 0.8329141 0.8025353
##           bfi_a    bfi_a6     bfi_c bfi_d_scale     bfi_e  bfi_hp8
## alpha 0.8117976 0.8028806 0.8218647   0.8056326 0.8749207 0.665913
##           bfi_n     bfi_o bfi_s_scale   hrz_col   hrz_ind       usi
## alpha 0.8745566 0.8100603   0.7937251 0.7041649 0.5645062 0.6852414
##         vrt_col   vrt_ind
## alpha 0.7266396 0.6538177
```

Generate lavaan syntax

This is an example of the measurement model diagram, using Horizontal Collectivism, for one group demonstrating constraints for the strict measurement invariance model. Paths with the same label are constrained to have the same path weight, and in the strict invariance model, all labeled path weights must be the same for each decade group. The residual covariances are an exception: the constraint imposed is that residual covariance is the same across the same time-lag, for the same indicator (so e.g., `cov(aind9, bind9) = cov(bind9, cind9)`, and `cov(aind9, cind9) = cov(bind9, dind9)`), but these are not constrained to be the same across group. Since this constraint is imposed in every factorial invariance model tested, it does not affect the fit comparisons.

```
## hrz_col_W1 ~ 0*1
## hrz_col_W1 ~~ 1*hrz_col_W1
## hrz_col_W1 =~ c(L1, L1, L1, L1)*aind9 + c(L2, L2, L2, L2)*aind10 + c(L3, L3, L3, L3)*aind11 + c(L4, L
## hrz_col_W2 =~ c(L1, L1, L1, L1)*bind9 + c(L2, L2, L2, L2)*bind10 + c(L3, L3, L3, L3)*bind11 + c(L4, L
## hrz_col_W3 =~ c(L1, L1, L1, L1)*cind9 + c(L2, L2, L2, L2)*cind10 + c(L3, L3, L3, L3)*cind11 + c(L4, L
## hrz_col_W4 =~ c(L1, L1, L1, L1)*dind9 + c(L2, L2, L2, L2)*dind10 + c(L3, L3, L3, L3)*dind11 + c(L4, L
##
## #---
##
## aind9 ~ c(int_1, int_1, int_1, int_1)*1
## bind9 ~ c(int_1, int_1, int_1, int_1)*1
## cind9 ~ c(int_1, int_1, int_1, int_1)*1
## dind9 ~ c(int_1, int_1, int_1, int_1)*1
##
## aind10 ~ c(int_2, int_2, int_2, int_2)*1
## bind10 ~ c(int_2, int_2, int_2, int_2)*1
## cind10 ~ c(int_2, int_2, int_2, int_2)*1
## dind10 ~ c(int_2, int_2, int_2, int_2)*1
##
## aind11 ~ c(int_3, int_3, int_3, int_3)*1
## bind11 ~ c(int_3, int_3, int_3, int_3)*1
## cind11 ~ c(int_3, int_3, int_3, int_3)*1
## dind11 ~ c(int_3, int_3, int_3, int_3)*1
##
```

```
## aind12 ~ c(int_4, int_4, int_4, int_4)*1
## bind12 ~ c(int_4, int_4, int_4, int_4)*1
## cind12 ~ c(int_4, int_4, int_4, int_4)*1
## dind12 ~ c(int_4, int_4, int_4, int_4)*1
##
## #---
##
## aind9 ~~ c(v_1, v_1, v_1, v_1)*aind9
## bind9 ~~ c(v_1, v_1, v_1, v_1)*bind9
## cind9 ~~ c(v_1, v_1, v_1, v_1)*cind9
## dind9 ~~ c(v_1, v_1, v_1, v_1)*dind9
##
## aind10 ~~ c(v_2, v_2, v_2, v_2)*aind10
## bind10 ~~ c(v_2, v_2, v_2, v_2)*bind10
## cind10 ~~ c(v_2, v_2, v_2, v_2)*cind10
## dind10 ~~ c(v_2, v_2, v_2, v_2)*dind10
##
## aind11 ~~ c(v_3, v_3, v_3, v_3)*aind11
## bind11 ~~ c(v_3, v_3, v_3, v_3)*bind11
## cind11 ~~ c(v_3, v_3, v_3, v_3)*cind11
## dind11 ~~ c(v_3, v_3, v_3, v_3)*dind11
##
## aind12 ~~ c(v_4, v_4, v_4, v_4)*aind12
## bind12 ~~ c(v_4, v_4, v_4, v_4)*bind12
## cind12 ~~ c(v_4, v_4, v_4, v_4)*cind12
## dind12 ~~ c(v_4, v_4, v_4, v_4)*dind12
##
## #---
##
## aind9 ~~ c(cv_11g1, cv_11g2, cv_11g3, cv_11g4)*bind9
## aind9 ~~ c(cv_12g1, cv_12g2, cv_12g3, cv_12g4)*cind9
## aind9 ~~ c(cv_13g1, cv_13g2, cv_13g3, cv_13g4)*dind9
## bind9 ~~ c(cv_11g1, cv_11g2, cv_11g3, cv_11g4)*cind9
## bind9 ~~ c(cv_12g1, cv_12g2, cv_12g3, cv_12g4)*dind9
## cind9 ~~ c(cv_11g1, cv_11g2, cv_11g3, cv_11g4)*dind9
##
## aind10 ~~ c(cv_21g1, cv_21g2, cv_21g3, cv_21g4)*bind10
## aind10 ~~ c(cv_22g1, cv_22g2, cv_22g3, cv_22g4)*cind10
## aind10 ~~ c(cv_23g1, cv_23g2, cv_23g3, cv_23g4)*dind10
## bind10 ~~ c(cv_21g1, cv_21g2, cv_21g3, cv_21g4)*cind10
## bind10 ~~ c(cv_22g1, cv_22g2, cv_22g3, cv_22g4)*dind10
## cind10 ~~ c(cv_21g1, cv_21g2, cv_21g3, cv_21g4)*dind10
##
## aind11 ~~ c(cv_31g1, cv_31g2, cv_31g3, cv_31g4)*bind11
## aind11 ~~ c(cv_32g1, cv_32g2, cv_32g3, cv_32g4)*cind11
## aind11 ~~ c(cv_33g1, cv_33g2, cv_33g3, cv_33g4)*dind11
## bind11 ~~ c(cv_31g1, cv_31g2, cv_31g3, cv_31g4)*cind11
## bind11 ~~ c(cv_32g1, cv_32g2, cv_32g3, cv_32g4)*dind11
## cind11 ~~ c(cv_31g1, cv_31g2, cv_31g3, cv_31g4)*dind11
##
## aind12 ~~ c(cv_41g1, cv_41g2, cv_41g3, cv_41g4)*bind12
## aind12 ~~ c(cv_42g1, cv_42g2, cv_42g3, cv_42g4)*cind12
## aind12 ~~ c(cv_43g1, cv_43g2, cv_43g3, cv_43g4)*dind12
## bind12 ~~ c(cv_41g1, cv_41g2, cv_41g3, cv_41g4)*cind12
```

```
## bind12 ~~ c(cv_42g1, cv_42g2, cv_42g3, cv_42g4)*dind12
## cind12 ~~ c(cv_41g1, cv_41g2, cv_41g3, cv_41g4)*dind12
```

```
library(semPlot)
semPaths(demo_model_fit, ask = F, include = 1, node.width = .75, layout = 'tree2', rotation = 2, curvatu
         intStyle = 'multi', levels = c(1,5,6,7), edge.color = '#bbbbbb', edge.label.color = '#444444',
         nCharEdges = 5, edge.label.cex = .5, nCharNodes = 8)
```

Results

MFI, CFI & *IC

To determine the invariance of measurement over groups, we can examine the χ^2 test. But this is often an overly strict test with a large sample size. Relative change in AIC and BIC (with lower values, and negative changes, being better) help guide interpretation by incorporating information about the number of parameters (AIC, BIC) and sample size (BIC). Cheung & Rensvold (2002) recomend using ΔCFI because it is not influenced by complexity or sample size, and does not correlate with overall fit measures. In this paper, they write, “A value of ΔCFI smaller than or equal to –0.01 indicates that the null hypothesis of invariance should not be rejected” (p 251). Below, I generally ignore the χ^2 tests in interpretation but present them for completeness.

```
eclude_cols <- c(3,5,6:9,12,14)
nothingburger <- do(group_by(invar_tests_and_models_df_w, factor_name),{
  table_df <- select(., -factor_name, -invar_type, -factor_name_c)
  title <- unique(.$factor_name_c)

  print(knitr::kable(
    table_df[,-eclude_cols],
    col.names = column_labels[-eclude_cols],
    caption = title,
    row.names = T))
  cat(paste0('\n\n<p>', conclusions_list[[title]], '</p>'))
  data.frame()
})
```

Table 1: Financial Aspirations

	Type	ΔAIC	ΔBIC	CFI	ΔCFI	ΔMFI	RMSEA
1	unconstrained			0.973			0.048 [0.041, 0.054]
2	long_metric			0.971	-0.002	-0.009	0.047 [0.040, 0.054]
3	long_strong			0.969	-0.002	-0.009	0.047 [0.040, 0.053]
4	long_strict			0.966	-0.003	-0.012	0.046 [0.040, 0.052]
5	baseline			0.959	-0.007	-0.032	0.049 [0.043, 0.055]
6	metric			0.958	-0.001	-0.004	0.049 [0.043, 0.055]
7	strong			0.956	-0.002	-0.011	0.050 [0.044, 0.056]
8	strict			0.954	-0.001	-0.006	0.050 [0.045, 0.056]

Invariant.

Table 2: Materialism

	Type	ΔAIC	ΔBIC	CFI	ΔCFI	ΔMFI	RMSEA
1	unconstrained						NA [NA, NA]
2	long_metric			0.729			0.068 [0.066, 0.070]
3	long_strong			0.727	-0.002	-0.001	0.067 [0.066, 0.069]
4	long_strict			0.727	0.000	0.000	0.066 [0.065, 0.068]
5	baseline			0.722	-0.005	-0.002	0.066 [0.064, 0.068]
6	metric			0.722	0.000	0.000	0.066 [0.064, 0.068]
7	strong			0.719	-0.003	-0.001	0.066 [0.064, 0.068]
8	strict			0.719	-0.001	-0.000	0.066 [0.064, 0.068]

Invariant.

Table 3: Horizontal Collectivism

	Type	ΔAIC	ΔBIC	CFI	ΔCFI	ΔMFI	RMSEA
1	unconstrained			0.973			0.045 [0.035, 0.054]
2	long_metric			0.969	-0.004	-0.009	0.045 [0.036, 0.054]
3	long_strong			0.962	-0.008	-0.019	0.048 [0.040, 0.056]
4	long_strict			0.960	-0.002	-0.004	0.046 [0.038, 0.054]
5	baseline			0.951	-0.009	-0.022	0.048 [0.041, 0.056]
6	metric			0.947	-0.004	-0.010	0.050 [0.043, 0.057]
7	strong			0.935	-0.012	-0.029	0.055 [0.048, 0.061]
8	strict			0.928	-0.007	-0.016	0.057 [0.050, 0.063]

Problematic: Neither CFI nor AIC support strict invariance. Constraints from metric to strong invariance (intercepts) may be an issue.

Table 4: Horizontal Individualism

	Type	ΔAIC	ΔBIC	CFI	ΔCFI	ΔMFI	RMSEA
1	unconstrained			0.942			0.053 [0.044, 0.061]
2	long_metric			0.944	0.001	0.002	0.049 [0.041, 0.058]
3	long_strong			0.941	-0.003	-0.005	0.048 [0.039, 0.056]
4	long_strict			0.935	-0.005	-0.009	0.047 [0.039, 0.055]
5	baseline			0.924	-0.011	-0.018	0.048 [0.041, 0.056]
6	metric			0.922	-0.002	-0.003	0.048 [0.041, 0.055]
7	strong			0.917	-0.005	-0.008	0.049 [0.042, 0.056]
8	strict			0.916	-0.001	-0.001	0.049 [0.042, 0.056]

Invariant.

Table 5: Mature Values Index

	Type	ΔAIC	ΔBIC	CFI	ΔCFI	ΔMFI	RMSEA
1	unconstrained			0.599			0.090 [0.088, 0.091]
2	long_metric			0.597	-0.002	-0.000	0.089 [0.087, 0.090]
3	long_strong			0.596	-0.002	-0.000	0.088 [0.087, 0.089]
4	long_strict			0.593	-0.002	-0.000	0.087 [0.086, 0.089]

	Type	ΔAIC	ΔBIC	CFI	ΔCFI	ΔMFI	RMSEA
5	baseline			0.588	-0.006	-0.000	0.087 [0.085, 0.088]
6	metric			0.587	-0.001	-0.000	0.087 [0.085, 0.088]
7	strong			0.582	-0.005	-0.000	0.087 [0.086, 0.088]
8	strict			0.579	-0.003	-0.000	0.087 [0.086, 0.088]

Likely invariant: Change in CFI and BIC suggest invariance, though AIC does not favor constraints beyond metric invariance.

Table 6: Unmitigated Self-Interest

	Type	ΔAIC	ΔBIC	CFI	ΔCFI	ΔMFI	RMSEA
1	unconstrained						NA [NA, NA]
2	long_metric			0.876			0.065 [0.061, 0.070]
3	long_strong			0.872	-0.004	-0.008	0.064 [0.060, 0.069]
4	long_strict			0.868	-0.004	-0.010	0.063 [0.058, 0.067]
5	baseline			0.862	-0.006	-0.013	0.062 [0.058, 0.066]
6	metric			0.860	-0.002	-0.005	0.062 [0.058, 0.066]
7	strong			0.853	-0.008	-0.017	0.063 [0.059, 0.067]
8	strict			0.846	-0.007	-0.015	0.064 [0.060, 0.068]

Possibly invariant: At each step, change in CFI suggests invariance, but baseline to strict change is greater than the recommended cutoff. Change in BIC suggests invariance, though AIC does not favor constraints beyond metric invariance.

Table 7: Vertical Collectivism

	Type	ΔAIC	ΔBIC	CFI	ΔCFI	ΔMFI	RMSEA
1	unconstrained			0.988			0.029 [0.014, 0.041]
2	long_metric			0.991	0.002	0.006	0.025 [0.003, 0.037]
3	long_strong			0.990	-0.000	-0.001	0.024 [0.000, 0.035]
4	long_strict			0.988	-0.003	-0.007	0.025 [0.009, 0.036]
5	baseline			0.978	-0.010	-0.025	0.032 [0.022, 0.041]
6	metric			0.973	-0.005	-0.012	0.035 [0.025, 0.043]
7	strong			0.963	-0.010	-0.025	0.040 [0.032, 0.048]
8	strict			0.961	-0.002	-0.006	0.041 [0.033, 0.049]

Possibly invariant: At each step, change in CFI suggests invariance, but baseline to strict change is greater than the recommended cutoff. Change in BIC suggests invariance, though AIC does not favor constraints except from strong to strict.

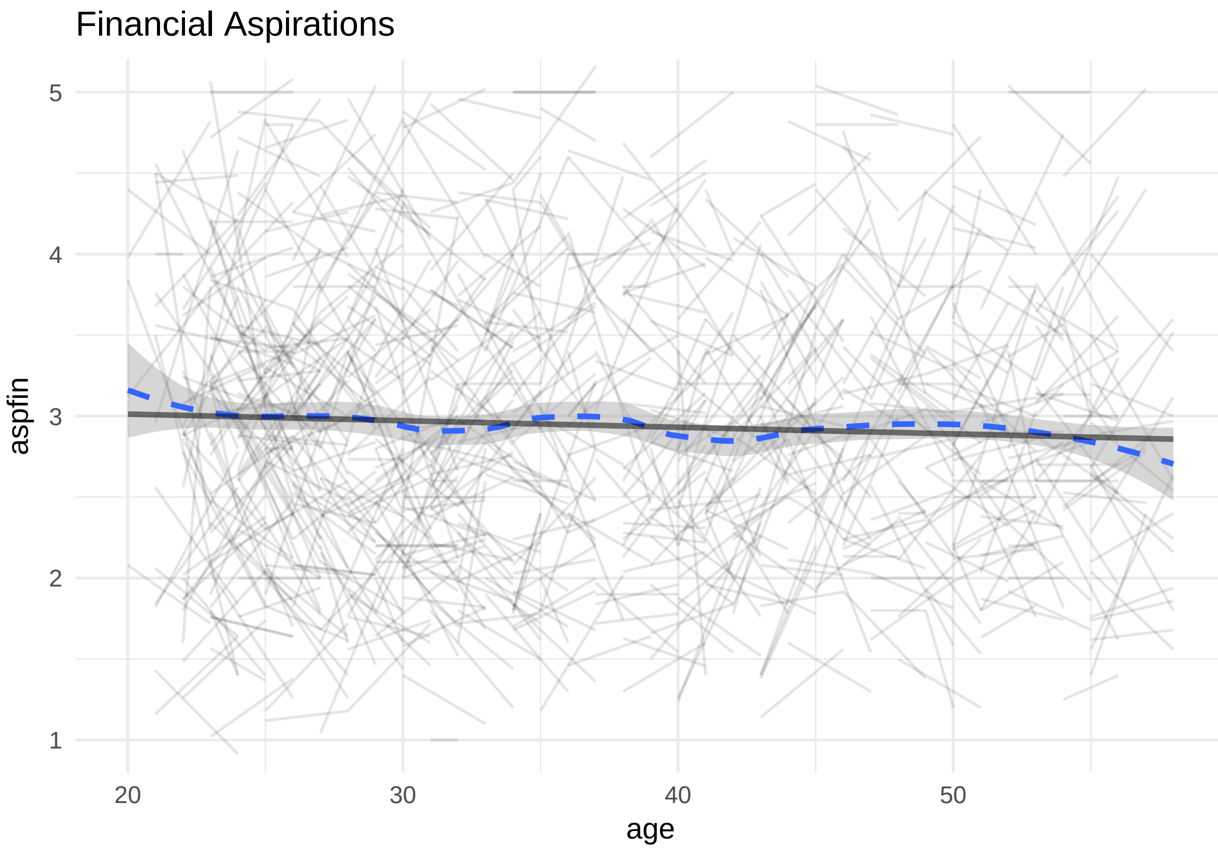
Table 8: Vertical Individualism

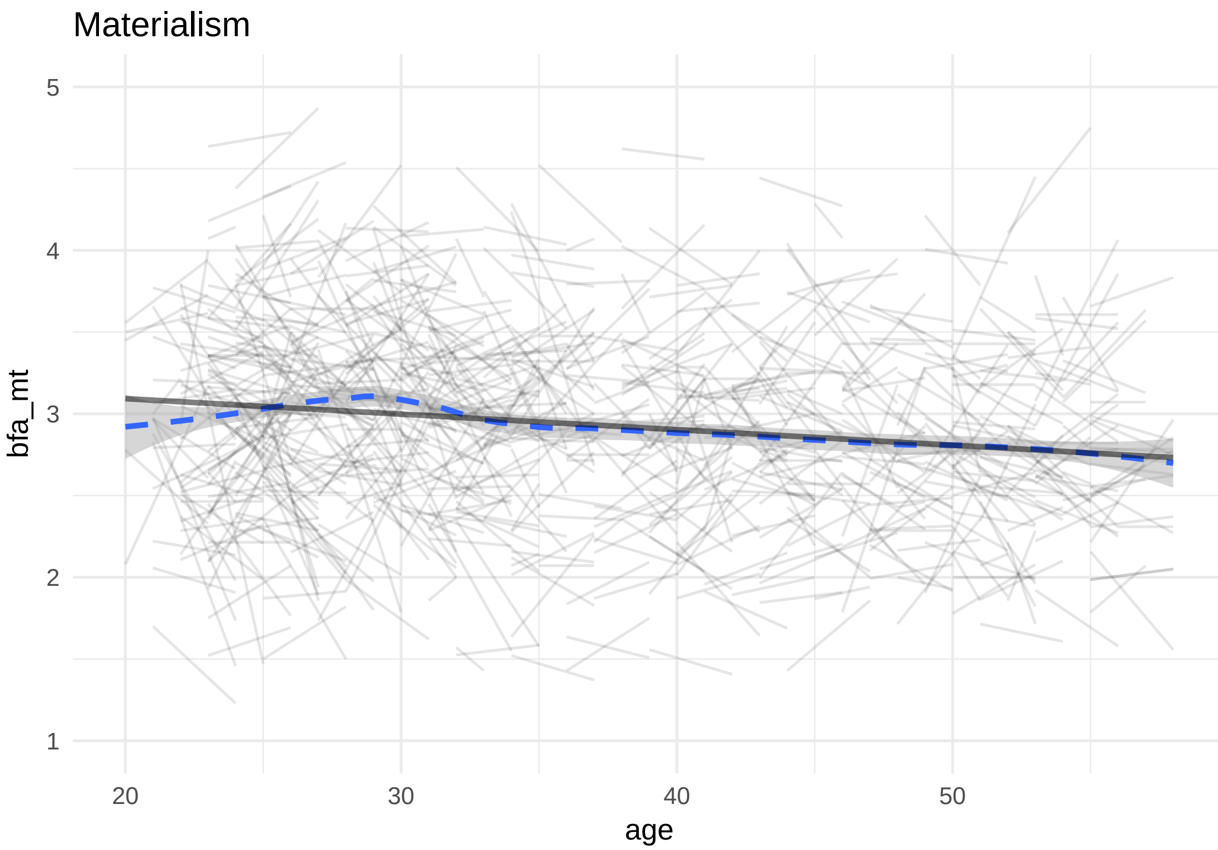
	Type	ΔAIC	ΔBIC	CFI	ΔCFI	ΔMFI	RMSEA
1	unconstrained			0.989			0.027 [0.009, 0.039]
2	long_metric			0.989	-0.000	-0.000	0.026 [0.007, 0.037]
3	long_strong			0.990	0.001	0.002	0.024 [0.000, 0.035]
4	long_strict			0.988	-0.002	-0.004	0.024 [0.006, 0.035]
5	baseline			0.978	-0.010	-0.025	0.031 [0.020, 0.040]
6	metric			0.977	-0.002	-0.004	0.032 [0.021, 0.041]

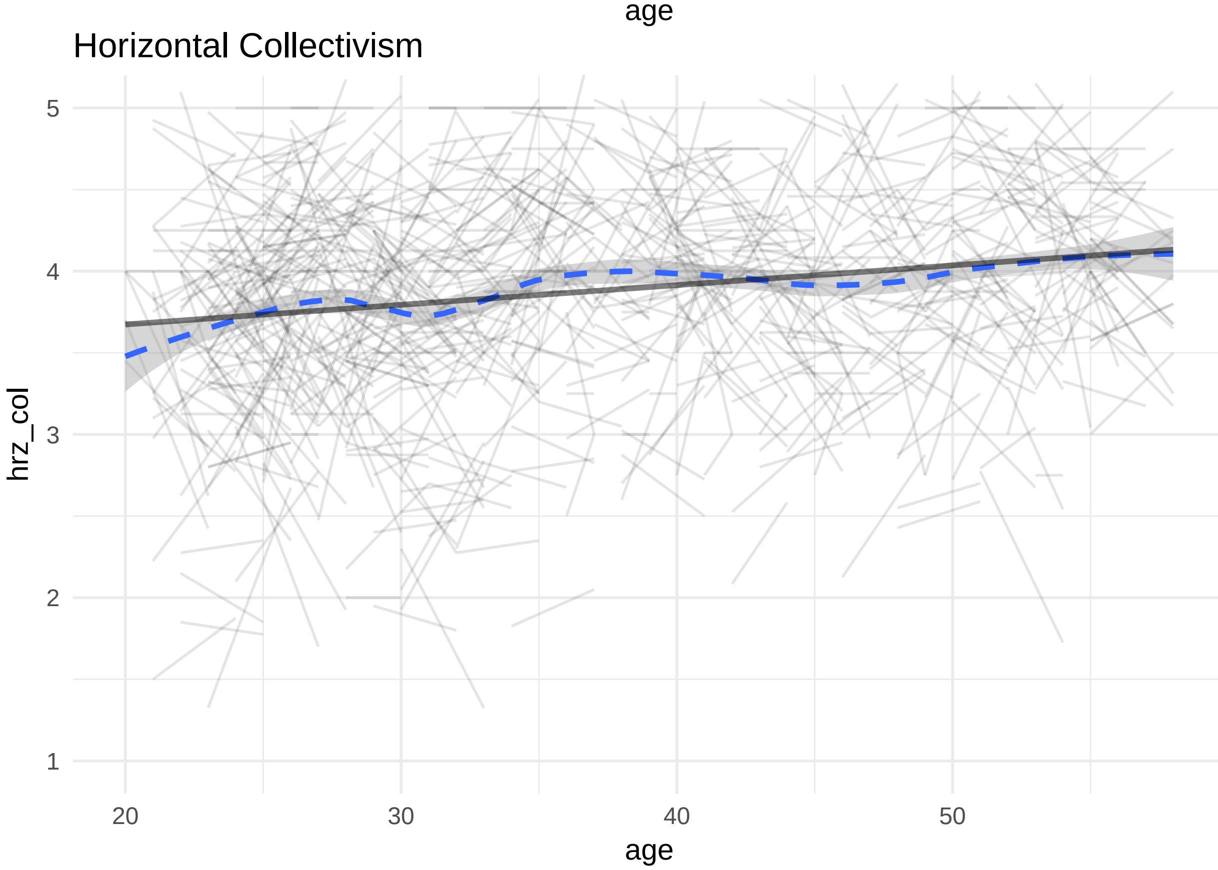
	Type	ΔAIC	ΔBIC	CFI	ΔCFI	ΔMFI	RMSEA
7	strong			0.965	-0.011	-0.027	0.038 [0.030, 0.046]
8	strict			0.965	-0.000	-0.001	0.038 [0.029, 0.046]

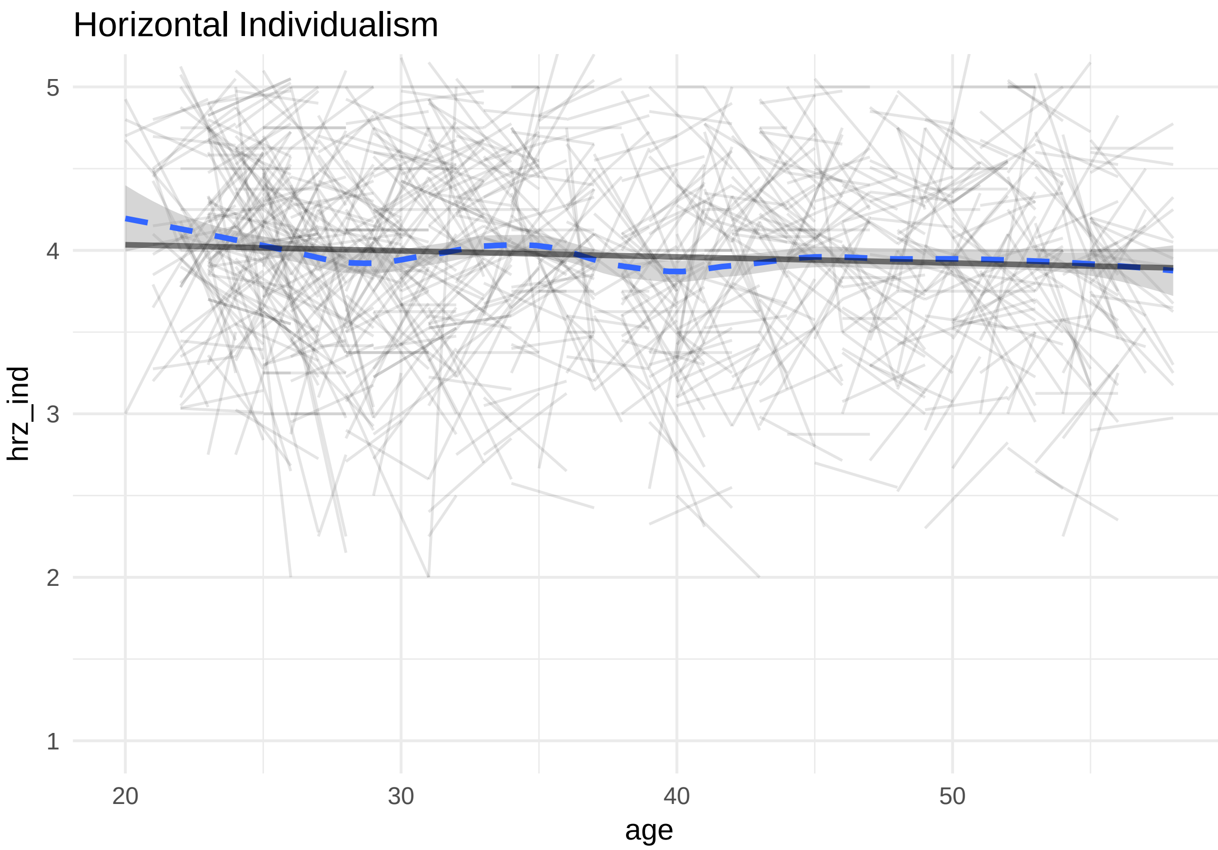
Possibly problematic: change in CFI rejects invariance between metric and strong, and between baseline and strict, which agrees with change in AIC. Change in BIC suggests invariance.

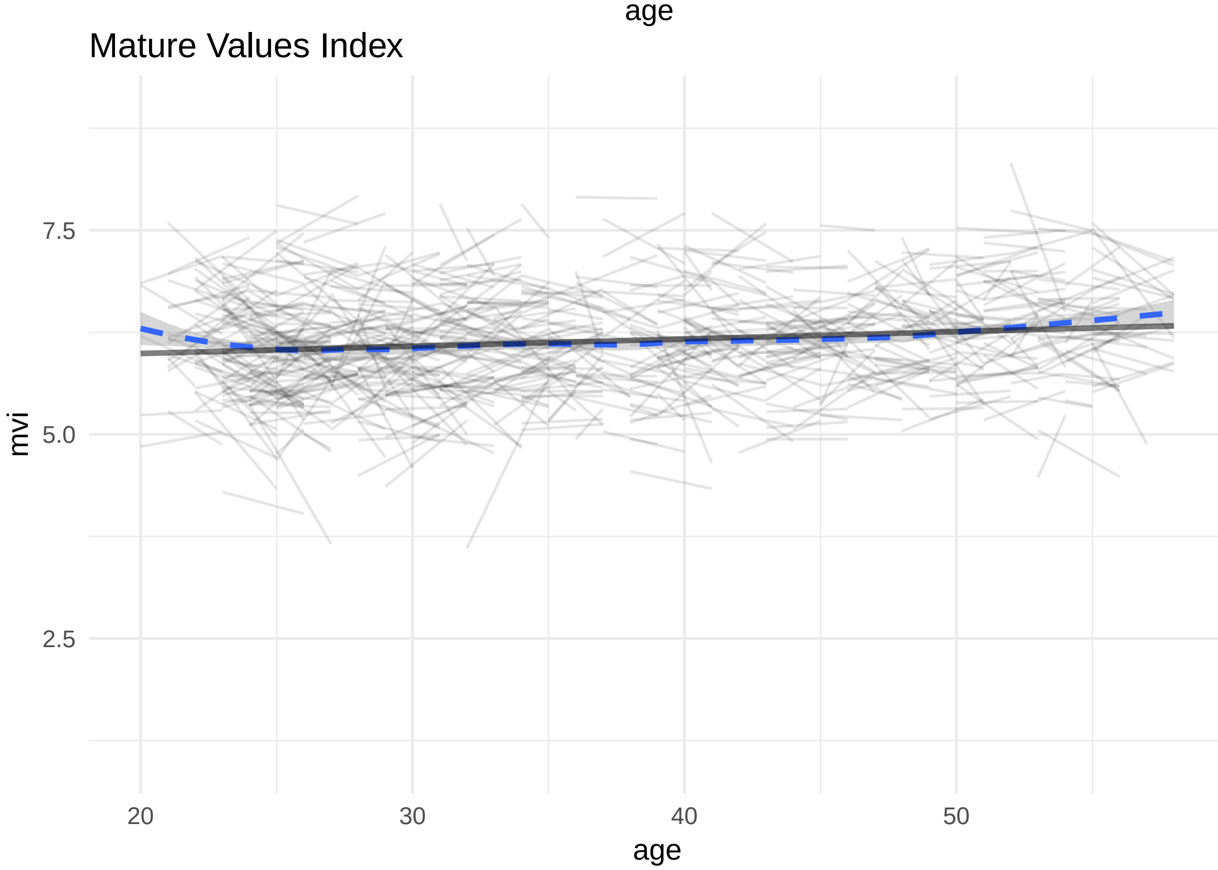
LOESS Plots



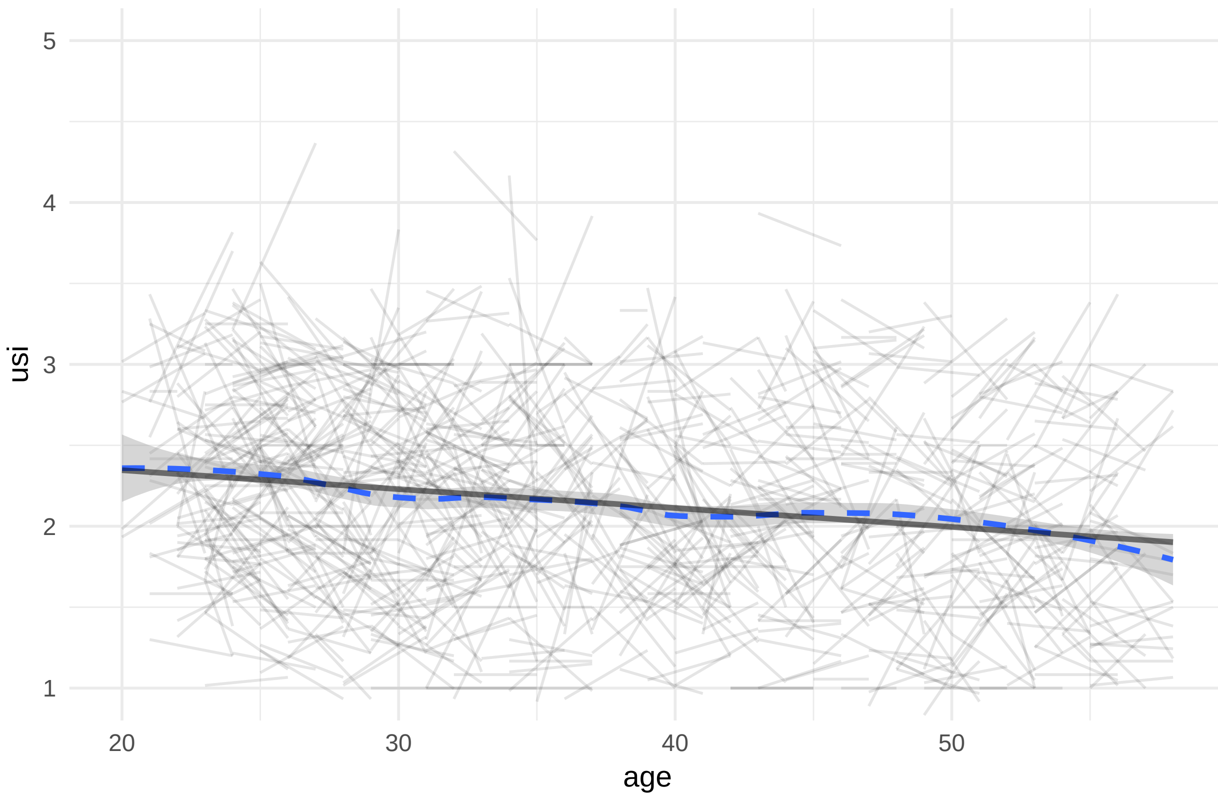

Materialism
5
4
3
2
1
bfa_mt
20
30
40
50
age


Horizontal Collectivism
5
4
3
2
1
hrz_col
20
30
40
50
age

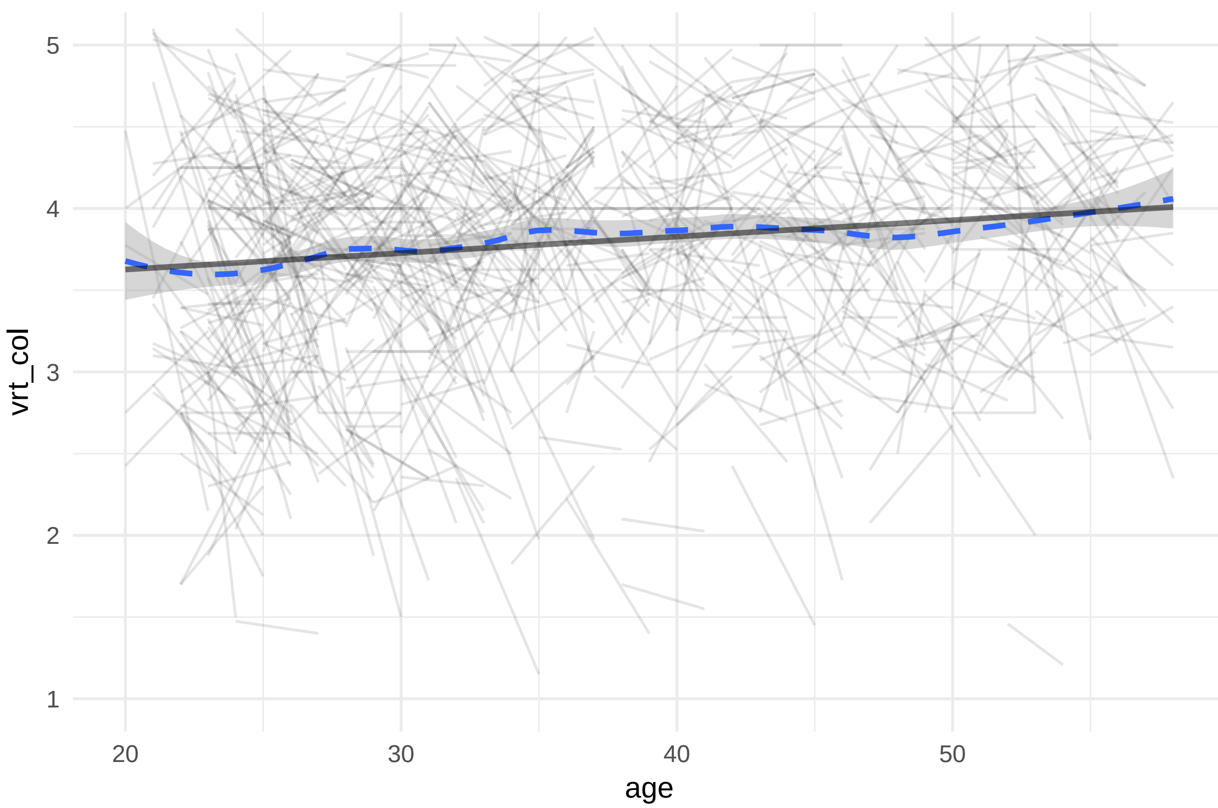

Horizontal Individualism
hrz_ind
5
4
3
2
1
20
30
40
50
age

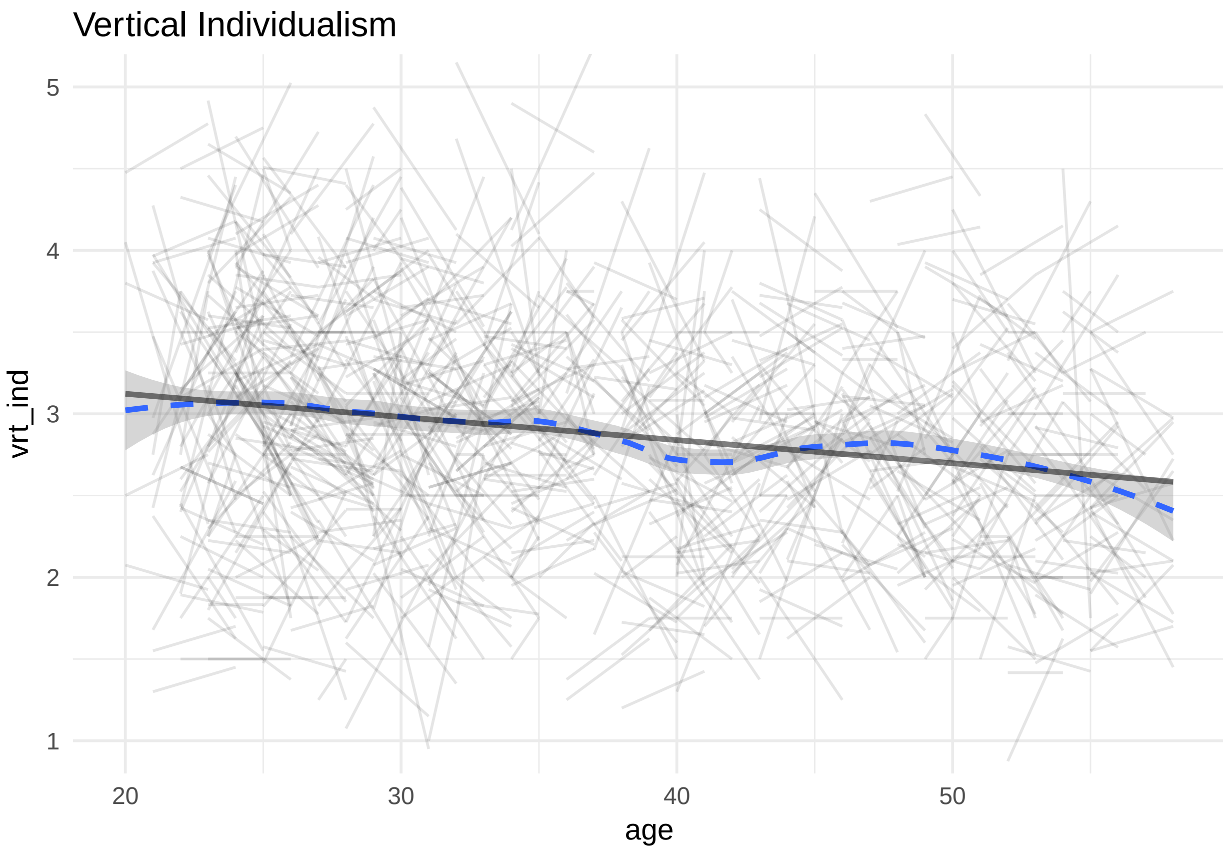

Mature Values Index
mvi
7.5
5.0
2.5
20
30
40
50
age

Unmitigated Self−Interest



Vertical Collectivism




Vertical Individualism
vrt_ind
5
4
3
2
1
20
30
40
50
age