Analysis 2019

Contents

Preparations

Load packages (see package information at the very end of this document)

```
library(here)
library(dplyr)
library(labelled)
library(ggplot2)
library(tidyr)
library(stringr)
library(psych)
library(lavaan)
library(semTools)
library(semPlot)
library(haven)
library(sjlabelled)
library(ggrepel)
library(ggpubr)
```

Read data file

```
df2019 <- readRDS("data/final/candsurvey_vaa_2019.rds")
```

Select variables used in the analysis and make sure the variable names are correct

```
VAA_LR_items<-c("h26","h27","h25","h28","y19")
VAA_LR_items %in% names(df2019)
```

```
## [1] TRUE TRUE TRUE TRUE TRUE
```

```
VAA_GT_items<-c("h21","h22","h13","h29","h24","y25")
VAA_GT_items %in% names(df2019)
```

```
## [1] TRUE TRUE TRUE TRUE TRUE TRUE
```

```
CS_LR_items<-c("C2b","C2g","C2h")
CS_LR_items %in% names(df2019)
```

```
## [1] TRUE TRUE TRUE
```

```
CS_GT_items<-c("C2a","C2c","C2d","C2e","C2f","C2i","C2j")
CS_GT_items %in% names(df2019)
```

```
## [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

```
#LR Self-placement
CS_LR_SP<-c("C5a")
CS_LR_SP %in% names(df2019)
```

```
## [1] TRUE
```

```
#LR imagined voter placement
CS_LR_IP<-c("C5c")
CS_LR_IP %in% names(df2019)
```

```
## [1] TRUE
```

```
Party_item<-c("puolue")
Party_item %in% names(df2019)
```

```
## [1] TRUE
```

```r
#vector for all item names

all_items<-c(Party_item,
             VAA_LR_items,
             VAA_GT_items,
             CS_LR_items,
             CS_GT_items,
             CS_LR_SP,
             CS_LR_IP)

#vector for observed variables in CFA (and party)

obs_items<-c(Party_item,
             VAA_LR_items,
             VAA_GT_items,
             CS_LR_items,
             CS_GT_items)

#a vector for indicator variables

ind_items<-c(
             VAA_LR_items,
             VAA_GT_items,
             CS_LR_items,
             CS_GT_items)
```

Print the responses to the observed items

```r
for (i in 1:length(obs_items)){
  print(obs_items[i])
  print(table(df2019[,obs_items[i]],useNA="always"))
  }
```

```
## [1] "puolue"
##
##   EOP    FP    IP    KD  KESK   KOK    KP   KTP  LIBE    LN  Muut   PIR    PS   RKP   SDP   SIN
##    17    38    47   190   216   211    50    32    36   108    28    87   213    98   216   152
##   SKE   SKP   STL   VAS  VIHR  <NA>
##    34    88   175   216   216     0
## [1] "h26"
##
##     1     2     3     4     5  <NA>
##   193   469    91   724   569   422
## [1] "h27"
##
##     1     2     3     4     5  <NA>
##   643   573    49   635   145   423
## [1] "h25"
##
##     1     2     3     4     5  <NA>
##   909   722    40   330    45   422
## [1] "h28"
##
##     1     2     3     4     5  <NA>
##   535   582    48   654   227   422
```

```
## [1] "y19"
##
##    1    2    4    5 <NA>
##   37  254  796 1172  209
## [1] "h21"
##
##    1    2    3    4    5 <NA>
##  281  251  106  281 1127  422
## [1] "h22"
##
##    1    2    3    4    5 <NA>
##  453  354  130  559  550  422
## [1] "h13"
##
##    1    2    3    4    5 <NA>
##  272  307   82  619  766  422
## [1] "h29"
##
##    1    2    3    4    5 <NA>
##  744  703   93  418   88  422
## [1] "h24"
##
##    1    2    3    4    5 <NA>
##  380  421   60  558  627  422
## [1] "y25"
##
##    1    2    4    5 <NA>
##  453  700  645  419  251
## [1] "C2b"
##
##    1    2    3    4    5 <NA>
##  250  314   59  101   24 1720
## [1] "C2g"
##
##    1    2    3    4    5 <NA>
##   29   97   84  299  242 1717
## [1] "C2h"
##
##    1    2    3    4    5 <NA>
##   48   94   77  220  313 1716
## [1] "C2a"
##
##    1    2    3    4    5 <NA>
##   15   49   69  324  294 1717
## [1] "C2c"
##
##    1    2    3    4    5 <NA>
##   36   96   94  229  298 1715
## [1] "C2d"
##
##    1    2    3    4    5 <NA>
##  461   79   74   51   86 1717
## [1] "C2e"
##
```

```
##    1    2    3    4    5 <NA>
##  183  164  249  125   31 1716
## [1] "C2f"
##
##    1    2    3    4    5 <NA>
##   37  142  156  267  148 1718
## [1] "C2i"
##
##    1    2    3    4    5 <NA>
##   87  103  127  277  158 1716
## [1] "C2j"
##
##    1    2    3    4    5 <NA>
##   49   64   72  144  424 1715
```

Data looks as it should!

Exclude completely missing cases

```
df2019$completely_missing<-
  rowSums(is.na(df2019[,ind_items]))==length(ind_items)

#number of completely missing cases
table(df2019$completely_missing)
```

```
##
## FALSE  TRUE
##  2365   103
```

```
#proportion of completely missing cases
100*table(df2019$completely_missing)/nrow(df2019)
```

```
##
##    FALSE     TRUE
## 95.82658  4.17342
```

```
#filter the used sample
dat2019<-df2019 %>%
  filter(!completely_missing)
```

Transform/Reverse code high scores on observed variable to indicate right and TAN positioning

```
reverse_items<-c("h26","y19",
                 "h21","h22","h13",
                 "C2g","C2h",
                 "C2c","C2e","C2i","C2j")

reverse_items %in% names(dat2019)
```

```
##  [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

```
for (i in 1:length(reverse_items)){
  dat2019[,reverse_items[i]]<-6-dat2019[,reverse_items[i]]
}
```

Analysis

Descriptive statistics

```
#look what parties there are
cbind(n=table(dat2019$puolue),
      proportion=round(100*prop.table(table(dat2019$puolue)),2))
```

```
##        n proportion
## EOP   17       0.72
## FP    38       1.61
## IP    41       1.73
## KD   188       7.95
## KESK 213       9.01
## KOK  211       8.92
## KP    46       1.95
## KTP   18       0.76
## LIBE  36       1.52
## LN   107       4.52
## Muut  23       0.97
## PIR   79       3.34
## PS   212       8.96
## RKP   97       4.10
## SDP  213       9.01
## SIN  132       5.58
## SKE   33       1.40
## SKP   79       3.34
## STL  157       6.64
## VAS  211       8.92
## VIHR 214       9.05
```

```
#how many responded to VAAs (any)

table(rowSums(is.na(dat2019[,c(VAA_LR_items,VAA_GT_items)]))!=
        length(c(VAA_LR_items,VAA_GT_items)))
```

```
##
## FALSE  TRUE
##    45  2320
```

```
#how many responded to CS (any)

table(rowSums(is.na(dat2019[,c(CS_LR_items,CS_GT_items)]))!=
        length(c(CS_LR_items,CS_GT_items)))
```

```
##
## FALSE  TRUE
##  1612   753
```

```
#table for CS-items

CS.item.table<-cbind.data.frame(
  item=c(CS_LR_items,CS_GT_items),
  description=c("The state should not interfere in economic activities",
                "Providing a stable social security network should be a state priority (r.)",
```

```r
                "The state should take measures to reduce income disparities (r.)",
                "Immigrants should adapt to Finnish habits",
                "Stronger measures should be taken to protect the environment (r.)",
                "Same Sex Marriages should be prohibited by law",
                "Women should be favored in job search and promotion (r.)",
                "People who break the law should be punished more severely",
                "Immigrants are good for the Finnish economy (r.)",
                "Deciding on abortion issues should be a women's right (r.)"),
  round(data.frame(describe(dat2019[,c(CS_LR_items,CS_GT_items)],fast=T))[,c("n","mean","sd")],2))

write.csv2(CS.item.table,"CS.item.table.csv")

#table for VAA-items

VAA.item.table<-cbind.data.frame(
  item=c(VAA_LR_items,VAA_GT_items),
  description=c("If there will be a situation where one is forced to either cut public services and soci
                "Large income inequalities are acceptable for compensating differences in people's talen
                "Public services should be outsourced more than they are now for private companies",
                "In the long run, the current extent of services and social benefits are too heavy for p
                "Public authorities should be the main provider of social and healthcare services (r.)"
                "Gay and lesbian couples should have the same marriage and adoption rights as straight c
                "If the government proposes to establish a refugee center in my home municipality, the p
                "For Finland, the advantages of the EU outweigh the disadvantages (r.)",
                "Economic growth and creation of jobs should be given primacy over environmental issues
                "Traditional values such as home, religion and fatherland form a good value base for pol
                "Finland must adopt tough measures to defend order and protect regular citizens"),
  round(data.frame(describe(dat2019[,c(VAA_LR_items,VAA_GT_items)],fast=T))[,c("n","mean","sd")],2))

write.csv2(VAA.item.table,"VAA.item.table.csv")
```

H1 and H2

H1. Left-Right placement as computed from responses to the pre-election public Voting Advice Applications (VAAs) is positively associated with Left-Right placement as computed from responses to the privately administered post-election Candidate Survey (CS). This association is stronger than any associations between the Left-Right and GAL-TAN dimensions.

H2. GAL-TAN placement as computed from responses to the pre-election public Voting Advice Applications (VAAs) is positively associated with GAL-TAN placement as computed from responses to the privately administered post-election Candidate Survey (CS). This association is stronger than any associations between the Left-Right and GAL-TAN dimensions.

Model script

```
model_H1H2<-"
#loadings
VAA_LR=~h26+h27+h25+h28+y19
VAA_GT=~h21+h22+h13+h29+h24+y25
CS_LR=~C2b+C2g+C2h
CS_GT=~C2a+C2c+C2d+C2e+C2f+C2i+C2j

#latent correlations

#cross-dimension same-method
VAA_LR~~r.VAA*VAA_GT
CS_LR~~r.CS*CS_GT

#concurrent validity
VAA_LR~~r.LR*CS_LR
VAA_GT~~r.GT*CS_GT

#cross-dimension cross-method correlations
VAA_LR~~r.d1*CS_GT
VAA_GT~~r.d2*CS_LR

#custom parameters
test.H1:=r.LR-max(r.VAA,r.CS,r.d1,r.d2)
test.H2:=r.GT-max(r.VAA,r.CS,r.d1,r.d2)

"
```

Fitting the model

```
fit_H1H2<-cfa(model=model_H1H2,
              data=dat2019,
              missing="fiml")
```

```
## Warning in lav_object_post_check(object): lavaan WARNING: covariance matrix of latent variables
##                   is not positive definite;
##                   use lavInspect(fit, "cov.lv") to investigate.
```

Some problems with latent variable covariance structure

```r
lavInspect(fit_H1H2, "cov.lv")
```

```
##        VAA_LR VAA_GT CS_LR CS_GT
## VAA_LR 1.038
## VAA_GT 0.475  1.222
## CS_LR  0.475  0.189  0.249
## CS_GT  0.237  0.677  0.105 0.343
```

```r
#examine standardized estimates
std.est_H1H2<-standardizedsolution(fit_H1H2)
std.est_H1H2[std.est_H1H2$op=="~~" &
               std.est_H1H2$lhs!=std.est_H1H2$rhs,]
```

```
##       lhs op    rhs est.std    se       z pvalue ci.lower ci.upper
## 22 VAA_LR ~~ VAA_GT   0.422 0.023  18.686      0    0.377    0.466
## 23  CS_LR ~~  CS_GT   0.360 0.037   9.725      0    0.288    0.433
## 24 VAA_LR ~~  CS_LR   0.934 0.020  45.885      0    0.894    0.973
## 25 VAA_GT ~~  CS_GT   1.045 0.010 101.601      0    1.025    1.065
## 26 VAA_LR ~~  CS_GT   0.397 0.029  13.540      0    0.339    0.454
## 27 VAA_GT ~~  CS_LR   0.342 0.035   9.704      0    0.273    0.411
```

There is an impossible correlation between GAL-TAN latent variables (absolute value > 1)

Respecified model: introduce the three preregistered residual correlations

Add the terms to the model script

```
model_H1H2.re<-paste0(model_H1H2,
                      "h27~~C2h\n",
                      "h21~~C2d\n",
                      "h29~~C2c\n")
```

Fitting the respecified model

```
fit_H1H2.re<-cfa(model=model_H1H2.re,
              data=dat2019,
              missing="fiml")
```

Results

Inspect fit of the model (first is the original model with problems, second is the respecified)

```
round(inspect(fit_H1H2,"fit")
      [c("npar","df","chisq","pvalue","cfi","tli","rmsea","srmr")],3)
```

```
##     npar       df    chisq   pvalue      cfi      tli    rmsea     srmr
##   69.000  183.000 2090.910    0.000    0.847    0.824    0.066    0.080
```

```
round(inspect(fit_H1H2.re,"fit")
      [c("npar","df","chisq","pvalue","cfi","tli","rmsea","srmr")],3)
```

```
##     npar       df    chisq   pvalue      cfi      tli    rmsea     srmr
##   72.000  180.000 1743.580    0.000    0.874    0.853    0.061    0.076
```

The fit of the model is adequate.

Hypotheses 1 and 2

Print standardized estimates to test the difference between correlations

```
std.est_H1H2<-standardizedsolution(fit_H1H2.re)
std.est_H1H2[std.est_H1H2$op==":=" |
               std.est_H1H2$op=="~~" &
               std.est_H1H2$lhs!=std.est_H1H2$rhs,]
```

```
##        lhs op                         rhs est.std    se      z pvalue
## 22  VAA_LR ~~                      VAA_GT   0.424 0.022 18.855      0
## 23   CS_LR ~~                       CS_GT   0.355 0.037  9.604      0
## 24  VAA_LR ~~                       CS_LR   0.915 0.020 44.726      0
## 25  VAA_GT ~~                       CS_GT   0.990 0.010 96.968      0
## 26  VAA_LR ~~                       CS_GT   0.407 0.029 14.175      0
## 27  VAA_GT ~~                       CS_LR   0.339 0.035  9.680      0
## 28     h27 ~~                         C2h   0.283 0.053  5.353      0
## 29     h21 ~~                         C2d   0.661 0.024 27.725      0
## 30     h29 ~~                         C2c   0.272 0.040  6.849      0
## 81 test.H1 := r.LR-max(r.VAA,r.CS,r.d1,r.d2)   0.492 0.030 16.340      0
## 82 test.H2 := r.GT-max(r.VAA,r.CS,r.d1,r.d2)   0.566 0.025 23.080      0
##    ci.lower ci.upper
## 22    0.380    0.468
```

```
## 23    0.282    0.427
## 24    0.875    0.956
## 25    0.970    1.010
## 26    0.350    0.463
## 27    0.270    0.408
## 28    0.179    0.387
## 29    0.615    0.708
## 30    0.194    0.350
## 81    0.433    0.551
## 82    0.518    0.614
```

```r
#save to a file
write.csv2(std.est_H1H2[std.est_H1H2$op!="~1",c(1:5,7)],
           "std.est_H1H2.csv")
```

H1: There is very strong (.915, p < .001) correlation between VAA-LR and CS-LR, and it is notably stronger (difference in correlations .492, p < .001) than the strongest of correlations between different dimensions (.424 between VAA_LR and VAA_GT, p < .001)

H2: There is very strong (.990, p < .001) correlation between VAA-GT and CS-GT, and it is notably stronger (difference in correlations .566, p < .001) than the strongest of correlations between different dimensions (.424 between VAA_LR and VAA_GT, p < .001)

Exploratory analysis for H1 and H2: Seek misspecification to improve the overall model fit

Residual correlations

```
mis.rescor_H1H2<-miPowerFit(fit_H1H2.re,cor=.20)
mis.rescor_H1H2<-mis.rescor_H1H2[mis.rescor_H1H2$op=="~~" &
                                   mis.rescor_H1H2$lhs!=mis.rescor_H1H2$rhs,]
#see summary of the decisions
table(mis.rescor_H1H2$decision.pow)
```

```
##
##  EPC:M EPC:NM     NM
##      1     68    138
```

```
#there are 1 residual correlation that is a misspecification

rounded.vars<-c("mi","epc","target.epc",
                "std.epc","se.epc")

num.round<-function(var){
  var<-as.numeric(var)
  var<-round(var,2)
  return(var)
}

mis.rescor_H1H2[,rounded.vars]<-sapply(mis.rescor_H1H2[,rounded.vars],num.round)

printed.vars<-c("lhs","op","rhs","mi","epc","target.epc",
                "std.epc","std.target.epc","significant.mi",
                "high.power","decision.pow","se.epc")

#print the output

mis.rescor_H1H2 %>%
  filter(mis.rescor_H1H2$decision.pow=="M" |
                mis.rescor_H1H2$decision.pow=="EPC:M") %>%
  dplyr::select(all_of(printed.vars))
```

```
##   lhs op rhs     mi  epc target.epc std.epc std.target.epc significant.mi
## 1 h25 ~~ y19 313.89 0.31       0.23    0.27            0.2           TRUE
##   high.power decision.pow se.epc
## 1       TRUE        EPC:M   0.02
```

There was one misspecified residual correlation in VAA-LR, between h25. Public services should be outsourced more than they are now for private companies and y19. Public authorities should be the main provider of social and healthcare services (r.)

Exploratory respecification of the model

Add new parameter to the model script

```
model_H1H2.exp.re<-paste0(model_H1H2.re,
                      "h25~~y19")
```

Fitting the exploratory model

```
fit_H1H2.exp.re<-cfa(model=model_H1H2.exp.re,
              data=dat2019,
              missing="fiml")
```

Results from the exploratory model

```
round(inspect(fit_H1H2.re,"fit")
      [c("npar","df","chisq","pvalue","cfi","tli","rmsea","srmr")],3)
```

```
##     npar       df    chisq   pvalue      cfi      tli    rmsea     srmr
##   72.000  180.000 1743.580    0.000    0.874    0.853    0.061    0.076
```

```
round(inspect(fit_H1H2.exp.re,"fit")
      [c("npar","df","chisq","pvalue","cfi","tli","rmsea","srmr")],3)
```

```
##     npar       df    chisq   pvalue      cfi      tli    rmsea     srmr
##   73.000  179.000 1439.326    0.000    0.899    0.881    0.055    0.073
```

The fit of the model is improved by additional residual correlation.

Retest Hypotheses 1 and 2

Print standardized estimates to test the difference between correlations

```
std.est_H1H2.exp<-standardizedsolution(fit_H1H2.exp.re)
std.est_H1H2.exp[std.est_H1H2.exp$op==":=" |
               std.est_H1H2.exp$op=="~~" &
               std.est_H1H2.exp$lhs!=std.est_H1H2.exp$rhs,]
```

```
##        lhs op                            rhs est.std    se      z pvalue
## 22  VAA_LR ~~                         VAA_GT   0.470 0.022 21.658      0
## 23   CS_LR ~~                          CS_GT   0.366 0.036 10.031      0
## 24  VAA_LR ~~                          CS_LR   0.932 0.021 45.253      0
## 25  VAA_GT ~~                          CS_GT   0.990 0.010 97.117      0
## 26  VAA_LR ~~                          CS_GT   0.441 0.028 15.520      0
## 27  VAA_GT ~~                          CS_LR   0.353 0.034 10.254      0
## 28     h27 ~~                            C2h   0.237 0.056  4.266      0
## 29     h21 ~~                            C2d   0.662 0.024 27.808      0
## 30     h29 ~~                            C2c   0.273 0.040  6.876      0
## 31     h25 ~~                            y19   0.426 0.020 20.857      0
## 82 test.H1 := r.LR-max(r.VAA,r.CS,r.d1,r.d2)   0.462 0.030 15.375      0
## 83 test.H2 := r.GT-max(r.VAA,r.CS,r.d1,r.d2)   0.520 0.024 21.777      0
##    ci.lower ci.upper
## 22    0.428    0.513
## 23    0.294    0.437
## 24    0.891    0.972
## 25    0.970    1.010
## 26    0.386    0.497
## 27    0.285    0.420
## 28    0.128    0.346
## 29    0.615    0.708
## 30    0.195    0.351
## 31    0.386    0.466
```

```
## 82    0.403    0.520
## 83    0.473    0.566
```

```r
#save to a file
write.csv2(std.est_H1H2.exp[std.est_H1H2.exp$op!="~1",c(1:5,7)],
           "std.est_H1H2.exp.csv")
```

The results are virtually identical to those without the additional residual correlation.

H1.exp: There is very strong (.932, $p < .001$) correlation between VAA-LR and CS-LR, and it is notably stronger (difference in correlations .462, $p < .001$) than the strongest of correlations between different dimensions (.470 between VAA_LR and VAA_GT, $p < .001$)

H2.exp: There is very strong (.990, $p < .001$) correlation between VAA-GT and CS-GT, and it is notably stronger (difference in correlations .520, $p < .001$) than the strongest of correlations between different dimensions (.470 between VAA_LR and VAA_GT, $p < .001$)

H5

H5. Left-Right self-placement in the privately administered post-election Candidate Survey (CS) is positively associated with Left-Right as computed from responses to the pre-election public Voting Advice Applications (VAAs). This association is stronger than the association between placement of an imagined party voter in the privately administered post-election Candidate Survey (CS) and Left-Right as computed from responses to the pre-election public Voting Advice Applications (VAAs).

Add placement variables and their correlations with latent factors to the model used for H1 and H2

```
model_H5<-"
#loadings
VAA_LR=~h26+h27+h25+h28+y19
VAA_GT=~h21+h22+h13+h29+h24+y25
CS_LR=~C2b+C2g+C2h
CS_GT=~C2a+C2c+C2d+C2e+C2f+C2i+C2j

#latent correlations

#cross-dimension same-method
VAA_LR~~r.VAA*VAA_GT
CS_LR~~r.CS*CS_GT

#concurrent validity
VAA_LR~~r.LR*CS_LR
VAA_GT~~r.GT*CS_GT

#cross-dimension cross-method correlations
VAA_LR~~r.d1*CS_GT
VAA_GT~~r.d2*CS_LR

#residual correlations
h27~~C2h
h21~~C2d
h29~~C2c

#placement variables (defined as quasi-latent variables)

SP_LR=~C5a
IP_LR=~C5c

VAA_LR~~r.self.LR*SP_LR
VAA_LR~~r.ideal.LR*IP_LR

#custom parameters
test.H1:=r.LR-max(r.VAA,r.CS,r.d1,r.d2)
test.H2:=r.GT-max(r.VAA,r.CS,r.d1,r.d2)
test.H5:=r.self.LR-r.ideal.LR
"
```

Fit the model

```
fit_H5<-cfa(model=model_H5,
            data=dat2019,
            missing="fiml")
```

Results

Inspect fit of the model

```
round(inspect(fit_H5,"fit")
      [c("npar","df","chisq","pvalue","cfi","tli","rmsea","srmr")],3)
```

```
##     npar       df    chisq   pvalue      cfi      tli    rmsea     srmr
##   85.000  214.000 1876.855    0.000    0.883    0.861    0.057    0.076
```

The fit of the model is adequate.

Hypothesis 5

Print standardized estimates to test the difference between correlations

```
std.est_H5<-standardizedsolution(fit_H5)
std.est_H5[std.est_H5$op==":=" |
               std.est_H5$op=="~~" &
               std.est_H5$lhs!=std.est_H5$rhs,]
```

```
##         lhs op                          rhs est.std    se      z pvalue
## 22   VAA_LR ~~                       VAA_GT   0.427 0.022 19.111      0
## 23    CS_LR ~~                        CS_GT   0.353 0.037  9.567      0
## 24   VAA_LR ~~                        CS_LR   0.917 0.020 45.304      0
## 25   VAA_GT ~~                        CS_GT   0.990 0.010 96.735      0
## 26   VAA_LR ~~                        CS_GT   0.409 0.029 14.356      0
## 27   VAA_GT ~~                        CS_LR   0.338 0.035  9.708      0
## 28      h27 ~~                          C2h   0.278 0.052  5.349      0
## 29      h21 ~~                          C2d   0.659 0.024 27.382      0
## 30      h29 ~~                          C2c   0.274 0.040  6.921      0
## 33   VAA_LR ~~                        SP_LR   0.829 0.015 55.090      0
## 34   VAA_LR ~~                        IP_LR   0.739 0.020 37.659      0
## 64   VAA_GT ~~                        SP_LR   0.540 0.025 21.566      0
## 65   VAA_GT ~~                        IP_LR   0.497 0.028 17.840      0
## 66    CS_LR ~~                        SP_LR   0.753 0.022 34.247      0
## 67    CS_LR ~~                        IP_LR   0.645 0.026 25.199      0
## 68    CS_GT ~~                        SP_LR   0.528 0.027 19.680      0
## 69    CS_GT ~~                        IP_LR   0.494 0.029 17.106      0
## 70    SP_LR ~~                        IP_LR   0.828 0.011 76.807      0
## 100 test.H1 := r.LR-max(r.VAA,r.CS,r.d1,r.d2)   0.490 0.030 16.372      0
## 101 test.H2 := r.GT-max(r.VAA,r.CS,r.d1,r.d2)   0.563 0.024 23.020      0
## 102 test.H5 :=          r.self.LR-r.ideal.LR   0.090 0.016  5.475      0
##     ci.lower ci.upper
## 22     0.383    0.471
## 23     0.281    0.426
## 24     0.877    0.956
## 25     0.970    1.010
## 26     0.353    0.465
```

```
## 27      0.270    0.407
## 28      0.176    0.380
## 29      0.612    0.706
## 30      0.196    0.351
## 33      0.800    0.859
## 34      0.700    0.777
## 64      0.490    0.589
## 65      0.443    0.552
## 66      0.710    0.796
## 67      0.595    0.695
## 68      0.476    0.581
## 69      0.438    0.551
## 70      0.807    0.849
## 100     0.431    0.548
## 101     0.515    0.611
## 102     0.058    0.122
```

```r
#save to a file
write.csv2(std.est_H5[std.est_H5$op!="~1",c(1:5,7)],
           "std.est_H5.csv")
```

H5. The correlation between VAA_LR and CS Self-placement on LR is large (.829, $p < .001$) and larger than the association between VAA_LR and placement of imagined party voter (.739, $p < .001$; difference .090, $p < .001$)

Exploratory H5: Seek misspecifications

Residual correlations

```r
mis.rescor_H5<-miPowerFit(fit_H5,cor=.20)
mis.rescor_H5<-mis.rescor_H5[mis.rescor_H5$op=="~~" &
                                   mis.rescor_H5$lhs!=mis.rescor_H5$rhs,]
#see summary of the decisions
table(mis.rescor_H5$decision.pow)
```

```
##
##  EPC:M EPC:NM     NM
##      1     81    167
```

```r
#there are 1 residual correlation that is a misspecification

rounded.vars<-c("mi","epc","target.epc",
                "std.epc","se.epc")

num.round<-function(var){
  var<-as.numeric(var)
  var<-round(var,2)
  return(var)
}

mis.rescor_H5[,rounded.vars]<-sapply(mis.rescor_H5[,rounded.vars],num.round)

printed.vars<-c("lhs","op","rhs","mi","epc","target.epc",
                "std.epc","std.target.epc","significant.mi",
                "high.power","decision.pow","se.epc")
```

```
#print the output

mis.rescor_H5 %>%
  filter(mis.rescor_H5$decision.pow=="M" |
               mis.rescor_H5$decision.pow=="EPC:M") %>%
  dplyr::select(all_of(printed.vars))
```

```
##   lhs op rhs     mi  epc target.epc std.epc std.target.epc significant.mi
## 1 h25 ~~ y19 297.11 0.29       0.23    0.25            0.2           TRUE
##   high.power decision.pow se.epc
## 1       TRUE        EPC:M   0.02
```

There was one misspecified residual correlation in VAA-LR, between h25. Public services should be outsourced more than they are now for private companies and y19. Public authorities should be the main provider of social and healthcare services (r.)

Exploratory respecification of the model

```
model_H5.exp<-paste0(model_H5,
                     "h25~~y19")
```

```
fit_H5.exp<-cfa(model=model_H5.exp,
              data=dat2019,
              missing="fiml")
```

Inspect fit of the model

```
round(inspect(fit_H5,"fit")
      [c("npar","df","chisq","pvalue","cfi","tli","rmsea","srmr")],3)
```

```
##     npar       df    chisq   pvalue      cfi      tli    rmsea     srmr
##   85.000  214.000 1876.855    0.000    0.883    0.861    0.057    0.076
```

```
round(inspect(fit_H5.exp,"fit")
      [c("npar","df","chisq","pvalue","cfi","tli","rmsea","srmr")],3)
```

```
##     npar       df    chisq   pvalue      cfi      tli    rmsea     srmr
##   86.000  213.000 1582.463    0.000    0.903    0.885    0.052    0.073
```

The fit of the model is improved.

Retest Hypothesis 5

Print standardized estimates to test the difference between correlations

```
std.est_H5.exp<-standardizedsolution(fit_H5.exp)
std.est_H5.exp[std.est_H5.exp$op==":=" |
               std.est_H5.exp$op=="~~" &
               std.est_H5.exp$lhs!=std.est_H5.exp$rhs,]
```

```
##         lhs op                          rhs est.std    se      z pvalue
## 22   VAA_LR ~~                       VAA_GT   0.472 0.022 21.792      0
## 23    CS_LR ~~                        CS_GT   0.364 0.036  9.966      0
## 24   VAA_LR ~~                        CS_LR   0.931 0.020 45.491      0
## 25   VAA_GT ~~                        CS_GT   0.990 0.010 96.827      0
## 26   VAA_LR ~~                        CS_GT   0.443 0.028 15.650      0
## 27   VAA_GT ~~                        CS_LR   0.351 0.034 10.234      0
```

```
## 28       h27 ~~                           C2h   0.238 0.054  4.423       0
## 29       h21 ~~                           C2d   0.659 0.024 27.486       0
## 30       h29 ~~                           C2c   0.274 0.040  6.916       0
## 33    VAA_LR ~~                         SP_LR   0.845 0.015 56.713       0
## 34    VAA_LR ~~                         IP_LR   0.751 0.020 38.037       0
## 35       h25 ~~                           y19   0.418 0.021 20.345       0
## 65    VAA_GT ~~                         SP_LR   0.544 0.025 21.883       0
## 66    VAA_GT ~~                         IP_LR   0.502 0.028 18.073       0
## 67     CS_LR ~~                         SP_LR   0.756 0.022 34.778       0
## 68     CS_LR ~~                         IP_LR   0.648 0.025 25.565       0
## 69     CS_GT ~~                         SP_LR   0.532 0.027 19.916       0
## 70     CS_GT ~~                         IP_LR   0.498 0.029 17.287       0
## 71     SP_LR ~~                         IP_LR   0.830 0.011 77.269       0
## 101 test.H1 := r.LR-max(r.VAA,r.CS,r.d1,r.d2)   0.460 0.030 15.378       0
## 102 test.H2 := r.GT-max(r.VAA,r.CS,r.d1,r.d2)   0.518 0.024 21.741       0
## 103 test.H5 :=           r.self.LR-r.ideal.LR   0.094 0.017  5.662       0
##     ci.lower ci.upper
## 22     0.429    0.514
## 23     0.292    0.435
## 24     0.891    0.972
## 25     0.970    1.010
## 26     0.387    0.498
## 27     0.284    0.419
## 28     0.133    0.344
## 29     0.612    0.706
## 30     0.196    0.351
## 33     0.816    0.874
## 34     0.712    0.790
## 35     0.378    0.458
## 65     0.495    0.593
## 66     0.448    0.556
## 67     0.713    0.799
## 68     0.599    0.698
## 69     0.480    0.585
## 70     0.442    0.554
## 71     0.809    0.851
## 101    0.401    0.518
## 102    0.471    0.565
## 103    0.061    0.126
```

```r
#save to a file
write.csv2(std.est_H5.exp[std.est_H5.exp$op!="~1",c(1:5,7)],
           "std.est_H5.exp.csv")
```

The results are virtually identical to those without the additional residual correlation.

H5.exp. The correlation between VAA_LR and CS Self-placement on LR is large (.845, $p < .001$) and larger than the association between VAA_LR and placement of imagined party voter (.751, $p < .001$; difference .094, $p < .001$)

H3 and H4

H3. Within-party placement on Left-Right as computed from responses to the pre-election public Voting Advice Applications (VAAs) is positively associated with within-party placement on Left-Right as computed from responses to the privately administered post-election Candidate Survey (CS). This association is stronger than any within-party associations between the Left-Right and GAL-TAN dimensions.

H4. Within-party placement on GAL-TAN as computed from responses to the pre-election public Voting Advice Applications (VAAs) is positively associated with within-party placement on GAL-TAN as computed from responses to the privately administered post-election Candidate Survey (CS). This association is stronger than any within-party associations between the Left-Right and GAL-TAN dimensions.

Construct a new dataframe that exclude other than members of the eight parties that have multiple members in the parliament

```
dat2019.party<-dat2019 %>%
  filter(puolue=="KD" |
           puolue=="KESK" |
           puolue=="KOK" |
           puolue=="PS" |
           puolue=="RKP" |
           puolue=="SDP" |
           puolue=="VAS" |
           puolue=="VIHR")

table(dat2019.party$puolue)
```

```
##
##   KD KESK  KOK   PS  RKP  SDP  VAS VIHR
##  188  213  211  212   97  213  211  214
```

Model script

Add names for group specific parameters

```
model_H3H4<-"
#loadings
VAA_LR=~h26+h27+h25+h28+y19
VAA_GT=~h21+h22+h13+h29+h24+y25
CS_LR=~C2b+C2g+C2h
CS_GT=~C2a+C2c+C2d+C2e+C2f+C2i+C2j

#latent correlations

#cross-dimension same-method
VAA_LR~~c(r.VAA.KD,r.VAA.KESK,r.VAA.KOK,r.VAA.PS,r.VAA.RKP,r.VAA.SDP,r.VAA.VAS,r.VAA.VIHR)*VAA_GT
CS_LR~~c(r.CS.KD,r.CS.KESK,r.CS.KOK,r.CS.PS,r.CS.RKP,r.CS.SDP,r.CS.VAS,r.CS.VIHR)*CS_GT

#concurrent validity
VAA_LR~~c(r.LR.KD,r.LR.KESK,r.LR.KOK,r.LR.PS,r.LR.RKP,r.LR.SDP,r.LR.VAS,r.LR.VIHR)*CS_LR
VAA_GT~~c(r.GT.KD,r.GT.KESK,r.GT.KOK,r.GT.PS,r.GT.RKP,r.GT.SDP,r.GT.VAS,r.GT.VIHR)*CS_GT

#cross-dimension cross-method correlations
VAA_LR~~c(r.d1.KD,r.d1.KESK,r.d1.KOK,r.d1.PS,r.d1.RKP,r.d1.SDP,r.d1.VAS,r.d1.VIHR)*CS_GT
VAA_GT~~c(r.d2.KD,r.d2.KESK,r.d2.KOK,r.d2.PS,r.d2.RKP,r.d2.SDP,r.d2.VAS,r.d2.VIHR)*CS_LR
```

```
#custom parameters
mean.r.VAA:=mean(r.VAA.KD,r.VAA.KESK,r.VAA.KOK,r.VAA.PS,r.VAA.RKP,r.VAA.SDP,r.VAA.VAS,r.VAA.VIHR)
mean.r.CS:=mean(r.CS.KD,r.CS.KESK,r.CS.KOK,r.CS.PS,r.CS.RKP,r.CS.SDP,r.CS.VAS,r.CS.VIHR)
mean.r.LR:=mean(r.LR.KD,r.LR.KESK,r.LR.KOK,r.LR.PS,r.LR.RKP,r.LR.SDP,r.LR.VAS,r.LR.VIHR)
mean.r.GT:=mean(r.GT.KD,r.GT.KESK,r.GT.KOK,r.GT.PS,r.GT.RKP,r.GT.SDP,r.GT.VAS,r.GT.VIHR)
mean.r.d1:=mean(r.d1.KD,r.d1.KESK,r.d1.KOK,r.d1.PS,r.d1.RKP,r.d1.SDP,r.d1.VAS,r.d1.VIHR)
mean.r.d2:=mean(r.d2.KD,r.d2.KESK,r.d2.KOK,r.d2.PS,r.d2.RKP,r.d2.SDP,r.d2.VAS,r.d2.VIHR)

test.H3:=mean.r.LR-max(mean.r.VAA,mean.r.CS,mean.r.d1,mean.r.d2)
test.H4:=mean.r.GT-max(mean.r.VAA,mean.r.CS,mean.r.d1,mean.r.d2)

"
```

Fit the configural model

```
fit_H3H4<-cfa(model=model_H3H4,
              data=dat2019.party,
              group=c("puolue"),
              group.label=c("KD","KESK","KOK","PS","RKP","SDP","VAS","VIHR"),
              missing="fiml")
```

```
## Warning in lav_model_estimate(lavmodel = lavmodel, lavpartable = lavpartable, : lavaan WARNING: the o
##                  but not all elements of the gradient are (near) zero;
##                  the optimizer may not have found a local solution
##                  use check.gradient = FALSE to skip this check.
```

Problems with finding a converging model. Add preregistered residual correlations.

Respecify model by adding the residual correlations

```
model_H3H4.re<-paste0(model_H3H4,
                      "h27~~C2h\n",
                      "h21~~C2d\n",
                      "h29~~C2c\n")
```

Fit the respecified model

```
fit_H3H4.re<-cfa(model=model_H3H4.re,
              data=dat2019.party,
              group=c("puolue"),
              group.label=c("KD","KESK","KOK","PS","RKP","SDP","VAS","VIHR"),
              missing="fiml")
```

```
## Warning in lav_model_estimate(lavmodel = lavmodel, lavpartable = lavpartable, : lavaan WARNING: the o
##                  but not all elements of the gradient are (near) zero;
##                  the optimizer may not have found a local solution
##                  use check.gradient = FALSE to skip this check.
```

The problem persists

Fit the model separately for each group

Note that here the model script for H1H2 must be used (it is identical to respecified H3H4 for single group)

```
fit_H3H4.re.KD<-cfa(model=model_H1H2.re,
                    data=dat2019.party,
                    group=c("puolue"),
                    group.label=c("KD"),
                    missing="fiml")
```

Model for KD Model for KD converges

```
round(inspect(fit_H3H4.re.KD,"fit")
      [c("npar","df","chisq","pvalue","cfi","tli","rmsea","srmr")],3)
```

```
##    npar      df   chisq  pvalue     cfi     tli   rmsea    srmr
##  72.000 180.000 295.998   0.000   0.705   0.656   0.059   0.122
```

```r
fit_H3H4.re.KESK<-cfa(model=model_H1H2.re,
                    data=dat2019.party,
                    group=c("puolue"),
                    group.label=c("KESK"),
                    missing="fiml")
```

Model for KESK

```
## Warning in lav_model_estimate(lavmodel = lavmodel, lavpartable = lavpartable, : lavaan WARNING: the o
##                     but not all elements of the gradient are (near) zero;
##                     the optimizer may not have found a local solution
##                     use check.gradient = FALSE to skip this check.
```

Model for KESK does not converge

```r
fit_H3H4.re.KOK<-cfa(model=model_H1H2.re,
                     data=dat2019.party,
                     group=c("puolue"),
                     group.label=c("KOK"),
                     missing="fiml")
```

Model for KOK   Model for KOK converges

```r
round(inspect(fit_H3H4.re.KOK,"fit")
      [c("npar","df","chisq","pvalue","cfi","tli","rmsea","srmr")],3)
```

```
##    npar      df   chisq  pvalue     cfi     tli   rmsea    srmr
##  72.000 180.000 310.526   0.000   0.632   0.571   0.059   0.137
```

```
fit_H3H4.re.PS<-cfa(model=model_H1H2.re,
                    data=dat2019.party,
                    group=c("puolue"),
                    group.label=c("PS"),
                    missing="fiml")
```

Model for PS

```
## Warning in lav_model_estimate(lavmodel = lavmodel, lavpartable = lavpartable, : lavaan WARNING: the o
##                    but not all elements of the gradient are (near) zero;
##                    the optimizer may not have found a local solution
##                    use check.gradient = FALSE to skip this check.
```

Model for PS does not converge

```r
fit_H3H4.re.RKP<-cfa(model=model_H1H2.re,
                    data=dat2019.party,
                    group=c("puolue"),
                    group.label=c("RKP"),
                    missing="fiml")
```

Model for RKP

```
## Warning in lav_object_post_check(object): lavaan WARNING: covariance matrix of latent variables
##                 is not positive definite;
##                 use lavInspect(fit, "cov.lv") to investigate.
```

Model for RKP converges, but has other problems

```r
round(inspect(fit_H3H4.re.RKP,"fit")
      [c("npar","df","chisq","pvalue","cfi","tli","rmsea","srmr")],3)
```

```
##    npar      df   chisq  pvalue     cfi     tli   rmsea    srmr
##  72.000 180.000 278.318   0.000   0.610   0.545   0.075   0.156
```

```r
fit_H3H4.re.SDP<-cfa(model=model_H1H2.re,
                    data=dat2019.party,
                    group=c("puolue"),
                    group.label=c("SDP"),
                    missing="fiml")
```

Model for SDP

```
## Warning in lav_object_post_check(object): lavaan WARNING: covariance matrix of latent variables
##                 is not positive definite;
##                 use lavInspect(fit, "cov.lv") to investigate.
```

Model for SDP conveges, but has other problems

```r
round(inspect(fit_H3H4.re.SDP,"fit")
      [c("npar","df","chisq","pvalue","cfi","tli","rmsea","srmr")],3)
```

```
##    npar      df   chisq  pvalue     cfi     tli   rmsea    srmr
##  72.000 180.000 382.844   0.000   0.633   0.572   0.073   0.127
```

```r
fit_H3H4.re.VAS<-cfa(model=model_H1H2.re,
                    data=dat2019.party,
                    group=c("puolue"),
                    group.label=c("VAS"),
                    missing="fiml")
```

Model for VAS

```
## Warning in lav_model_estimate(lavmodel = lavmodel, lavpartable = lavpartable, :
## lavaan WARNING: the optimizer warns that a solution has NOT been found!
```

Model for VAS does not converge

```r
fit_H3H4.re.VIHR<-cfa(model=model_H1H2.re,
                     data=dat2019.party,
                     group=c("puolue"),
                     group.label=c("VIHR"),
                     missing="fiml")
```

Model for VIHR

```
## Warning in lav_model_estimate(lavmodel = lavmodel, lavpartable = lavpartable, : lavaan WARNING: the o
##                     but not all elements of the gradient are (near) zero;
##                     the optimizer may not have found a local solution
##                     use check.gradient = FALSE to skip this check.
```

Model for VIHR does not converge

Summary of H3-H4 with MG-CFA approach

The configural model did not converge, even after respecification. Single group models also were non-converging or had other type of problems, except for KD and KOK, for which the fit of the model nevertheless was poor, and therefore not interpretable.

This most likely is an indication that the sample sizes of the parties are too small for this model with 21 indicators and 4 factors.

The alternative way to test hypotheses 4-6 is presented below. It unconfounds the associations in the model by using party-mean centered observed variables for estimating the similar type of model that was used for H1 and H2, and H5, respectively. Because this approach does not have any grouping structure, it uses the overall sample size for the eight parties, which is 1559. It is nevertheless only conducted among the eight focal parties, and other parties are excluded. Because the misspecification in the model with centered variables might be entirely different to raw score variables, the modeling is again started with no residual correlations and they are examined if the fit of the model is inadequate.

H3 and H4 with group-mean centered variables and no grouping structure

ICC: Estimate how much of the variation in each item is between-groups

```
#there was problems running the mult.icc function to the data structure so
#data observed data was extracted from one of the previously fitted models
#to get rid of all labels etc.
num.dat.2019<-data.frame(fit_H5@Data@X,dat2019$puolue)
names(num.dat.2019)<-c(fit_H5@Data@ov$name,"puolue")
num.dat.2019<-num.dat.2019 %>%
  filter(puolue=="KD" |
           puolue=="KESK" |
           puolue=="KOK" |
           puolue=="PS" |
           puolue=="RKP" |
           puolue=="SDP" |
           puolue=="VAS" |
           puolue=="VIHR")


ICC<-data.frame(
  multilevel::mult.icc(x=num.dat.2019[,
                                      all_items[2:length(all_items)]],
                       grpid=num.dat.2019$puolue))
ICC[,2:3]<-round(ICC[,2:3],3)
ICC
```

```
##    Variable  ICC1  ICC2
## 1       h26 0.484 0.995
## 2       h27 0.443 0.994
## 3       h25 0.483 0.995
## 4       h28 0.415 0.993
## 5       y19 0.354 0.991
## 6       h21 0.647 0.997
## 7       h22 0.490 0.995
## 8       h13 0.552 0.996
## 9       h29 0.327 0.990
## 10      h24 0.600 0.997
## 11      y25 0.345 0.990
## 12      C2b 0.127 0.966
## 13      C2g 0.251 0.985
## 14      C2h 0.444 0.994
## 15      C2a 0.295 0.988
## 16      C2c 0.405 0.993
## 17      C2d 0.501 0.995
## 18      C2e 0.103 0.957
## 19      C2f 0.213 0.981
## 20      C2i 0.515 0.995
## 21      C2j 0.419 0.993
## 22      C5a 0.683 0.998
## 23      C5c 0.767 0.998
```

```
describe(100*ICC$ICC1,fast=T)
```

```
##    vars  n  mean    sd  min  max range   se
```

```
## X1    1 23 42.88 16.67 10.3 76.7  66.4 3.48
```

```r
ICC$label<-get_label(df2019[,as.character(ICC[,1])])

#export to .csv file
write.csv2(ICC,"ICC_2019.csv")
```

ICC1 gives the proportion (%) of variance that is between the parties (ICC2 is the reliability of the group means). There is quite a lot of between-party variance, but the responses are not entire defined by party either.

Variable centering

```
dat2019.gmc<-data.frame(dat2019.party)
na.mean<-function(var){
  mean(var,na.rm=T)
}

group.means<-dat2019.gmc %>%
  group_by(puolue) %>%
  summarise_at(all_items[2:length(all_items)],na.mean)


dat2019.gmc<-left_join(x=dat2019.gmc,
                       y=group.means,
                       by=c("puolue"),
                       suffix=c("",".pm"))




for(i in all_items[2:length(all_items)]){
  dat2019.gmc[i]<-dat2019.gmc[,i]-dat2019.gmc[,which(grepl(i,names(dat2019.gmc)) &
                           grepl("pm",names(dat2019.gmc)) &
                     !grepl("r",names(dat2019.gmc))) ]
}

describe(dat2019.gmc[,all_items],fast=T)
```

```
## Warning in FUN(newX[, i], ...): no non-missing arguments to min; returning Inf

## Warning in FUN(newX[, i], ...): no non-missing arguments to max; returning -Inf

##        vars    n mean   sd   min  max range   se
## puolue    1 1559  NaN   NA   Inf -Inf  -Inf   NA
## h26       2 1425    0 0.97 -3.02 3.78  6.80 0.03
## h27       3 1424    0 1.03 -2.92 3.10  6.02 0.03
## h25       4 1425    0 0.81 -2.57 3.11  5.68 0.02
## h28       5 1425    0 1.08 -2.93 3.73  6.66 0.03
## y19       6 1528    0 0.82 -2.02 3.93  5.95 0.02
## h21       7 1425    0 0.94 -3.44 3.86  7.30 0.02
## h22       8 1425    0 1.08 -2.84 3.19  6.03 0.03
## h13       9 1425    0 0.88 -3.22 3.67  6.89 0.02
## h29      10 1425    0 0.99 -2.43 3.55  5.98 0.03
## h24      11 1425    0 0.97 -3.80 3.27  7.06 0.03
## y25      12 1504    0 1.16 -3.41 3.29  6.70 0.03
## C2b      13  475    0 0.94 -1.70 3.52  5.22 0.04
## C2g      14  476    0 0.94 -2.26 3.11  5.37 0.04
## C2h      15  476    0 0.92 -2.72 2.88  5.60 0.04
## C2a      16  477    0 0.78 -3.47 1.48  4.96 0.04
## C2c      17  477    0 0.91 -2.30 3.24  5.53 0.04
## C2d      18  475    0 0.96 -2.87 3.76  6.62 0.04
## C2e      19  477    0 1.04 -2.83 2.12  4.95 0.05
## C2f      20  477    0 1.01 -2.72 2.29  5.01 0.05
## C2i      21  477    0 0.89 -3.45 3.18  6.64 0.04
## C2j      22  477    0 0.95 -2.71 3.77  6.47 0.04
## C5a      23  473    0 1.51 -5.31 5.39 10.71 0.07
```

```
## C5c        24  470    0 1.07 -3.15 3.16  6.31 0.05
```

Define the model

Identical to the model for H1 and H2

```
model_H3H4<-"
#loadings
VAA_LR=~h26+h27+h25+h28+y19
VAA_GT=~h21+h22+h13+h29+h24+y25
CS_LR=~C2b+C2g+C2h
CS_GT=~C2a+C2c+C2d+C2e+C2f+C2i+C2j

#latent correlations

#cross-dimension same-method
VAA_LR~~r.VAA*VAA_GT
CS_LR~~r.CS*CS_GT

#concurrent validity
VAA_LR~~r.LR*CS_LR
VAA_GT~~r.GT*CS_GT

#cross-dimension cross-method correlations
VAA_LR~~r.d1*CS_GT
VAA_GT~~r.d2*CS_LR

#custom parameters
test.H3:=r.LR-max(r.VAA,r.CS,r.d1,r.d2)
test.H4:=r.GT-max(r.VAA,r.CS,r.d1,r.d2)

"
```

Fit the model

```
fit_H3H4<-cfa(model=model_H3H4,
              data=dat2019.gmc,
              missing="fiml")
```

```
## Warning in lav_object_post_check(object): lavaan WARNING: covariance matrix of latent variables
##                 is not positive definite;
##                 use lavInspect(fit, "cov.lv") to investigate.
```

Problems with latent variable covariance matrix

```
lavInspect(fit_H3H4, "cov.lv")
```

```
##        VAA_LR VAA_GT CS_LR CS_GT
## VAA_LR 0.290
## VAA_GT 0.039  0.167
## CS_LR  0.116  0.011  0.068
## CS_GT  0.034  0.157  0.016 0.127
```

```
summary(fit_H3H4,fit=T,standardized=T,rsquare=T)
```

```
## lavaan 0.6-5 ended normally after 89 iterations
##
```

```
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         69
##
##   Number of observations                          1559
##   Number of missing patterns                        17
##
## Model Test User Model:
##
##   Test statistic                               756.826
##   Degrees of freedom                               183
##   P-value (Chi-square)                           0.000
##
## Model Test Baseline Model:
##
##   Test statistic                              2537.225
##   Degrees of freedom                               210
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.753
##   Tucker-Lewis Index (TLI)                       0.717
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -27534.678
##   Loglikelihood unrestricted model (H1)             NA
##
##   Akaike (AIC)                               55207.356
##   Bayesian (BIC)                             55576.630
##   Sample-size adjusted Bayesian (BIC)        55357.432
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.045
##   90 Percent confidence interval - lower         0.042
##   90 Percent confidence interval - upper         0.048
##   P-value RMSEA <= 0.05                          0.995
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.069
##
## Parameter Estimates:
##
##   Information                                 Observed
##   Observed information based on                Hessian
##   Standard errors                             Standard
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   VAA_LR =~
##     h26               1.000                               0.539    0.556
```

```
##     h27               0.926    0.080   11.569    0.000    0.499    0.486
##     h25               0.835    0.071   11.697    0.000    0.450    0.553
##     h28               0.865    0.079   10.990    0.000    0.466    0.434
##     y19               0.628    0.063    9.964    0.000    0.339    0.413
##   VAA_GT =~
##     h21               1.000                               0.408    0.434
##     h22               1.319    0.138    9.534    0.000    0.539    0.499
##     h13               0.682    0.088    7.739    0.000    0.278    0.316
##     h29               0.950    0.108    8.822    0.000    0.388    0.390
##     h24               1.055    0.105   10.029    0.000    0.431    0.443
##     y25               1.450    0.148    9.777    0.000    0.592    0.509
##   CS_LR =~
##     C2b               1.000                               0.260    0.278
##     C2g               1.919    0.403    4.764    0.000    0.499    0.534
##     C2h               2.822    0.547    5.158    0.000    0.734    0.798
##   CS_GT =~
##     C2a               1.000                               0.357    0.458
##     C2c               1.002    0.176    5.687    0.000    0.357    0.393
##     C2d               1.062    0.188    5.639    0.000    0.378    0.392
##     C2e               0.449    0.168    2.674    0.007    0.160    0.154
##     C2f               1.000    0.174    5.753    0.000    0.357    0.352
##     C2i               1.374    0.193    7.107    0.000    0.490    0.547
##     C2j               0.549    0.161    3.406    0.001    0.196    0.205
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   VAA_LR ~~
##     VAA_GT  (r.VA)    0.039    0.010    3.788    0.000    0.179    0.179
##   CS_LR ~~
##     CS_GT   (r.CS)    0.016    0.008    2.084    0.037    0.172    0.172
##   VAA_LR ~~
##     CS_LR   (r.LR)    0.116    0.024    4.835    0.000    0.826    0.826
##   VAA_GT ~~
##     CS_GT   (r.GT)    0.157    0.022    7.219    0.000    1.079    1.079
##   VAA_LR ~~
##     CS_GT   (r.d1)    0.034    0.014    2.457    0.014    0.179    0.179
##   VAA_GT ~~
##     CS_LR   (r.d2)    0.011    0.008    1.361    0.173    0.099    0.099
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .h26              -0.001    0.026   -0.059    0.953   -0.001   -0.002
##    .h27              -0.002    0.027   -0.060    0.952   -0.002   -0.002
##    .h25              -0.001    0.021   -0.058    0.953   -0.001   -0.002
##    .h28              -0.001    0.028   -0.046    0.964   -0.001   -0.001
##    .y19              -0.002    0.021   -0.073    0.942   -0.002   -0.002
##    .h21              -0.000    0.025   -0.020    0.984   -0.000   -0.001
##    .h22              -0.001    0.028   -0.022    0.982   -0.001   -0.001
##    .h13              -0.000    0.023   -0.014    0.989   -0.000   -0.000
##    .h29              -0.000    0.026   -0.018    0.986   -0.000   -0.000
##    .h24              -0.001    0.026   -0.020    0.984   -0.001   -0.001
##    .y25              -0.001    0.030   -0.021    0.983   -0.001   -0.001
##    .C2b              -0.005    0.043   -0.116    0.908   -0.005   -0.005
##    .C2g              -0.010    0.041   -0.231    0.817   -0.010   -0.010
```

```
##    .C2h              -0.015    0.039   -0.378    0.706   -0.015   -0.016
##    .C2a               0.036    0.034    1.053    0.292    0.036    0.046
##    .C2c               0.036    0.040    0.893    0.372    0.036    0.040
##    .C2d               0.038    0.043    0.872    0.383    0.038    0.039
##    .C2e               0.016    0.048    0.338    0.735    0.016    0.016
##    .C2f               0.036    0.045    0.792    0.429    0.036    0.035
##    .C2i               0.049    0.038    1.287    0.198    0.049    0.055
##    .C2j               0.020    0.044    0.453    0.650    0.020    0.021
##     VAA_LR            0.000                               0.000    0.000
##     VAA_GT            0.000                               0.000    0.000
##     CS_LR             0.000                               0.000    0.000
##     CS_GT             0.000                               0.000    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .h26               0.650    0.033   19.975    0.000    0.650    0.691
##    .h27               0.806    0.037   21.684    0.000    0.806    0.764
##    .h25               0.459    0.024   18.968    0.000    0.459    0.694
##    .h28               0.939    0.041   23.083    0.000    0.939    0.812
##    .y19               0.557    0.024   23.245    0.000    0.557    0.829
##    .h21               0.717    0.032   22.480    0.000    0.717    0.811
##    .h22               0.874    0.041   21.363    0.000    0.874    0.751
##    .h13               0.699    0.028   24.964    0.000    0.699    0.900
##    .h29               0.839    0.035   23.964    0.000    0.839    0.848
##    .h24               0.759    0.033   22.744    0.000    0.759    0.804
##    .y25               1.003    0.047   21.363    0.000    1.003    0.741
##    .C2b               0.809    0.054   14.954    0.000    0.809    0.923
##    .C2g               0.625    0.049   12.708    0.000    0.625    0.715
##    .C2h               0.307    0.061    5.052    0.000    0.307    0.363
##    .C2a               0.479    0.034   13.899    0.000    0.479    0.790
##    .C2c               0.696    0.048   14.407    0.000    0.696    0.845
##    .C2d               0.790    0.056   14.162    0.000    0.790    0.847
##    .C2e               1.058    0.069   15.298    0.000    1.058    0.976
##    .C2f               0.901    0.062   14.584    0.000    0.901    0.876
##    .C2i               0.561    0.044   12.842    0.000    0.561    0.701
##    .C2j               0.872    0.057   15.170    0.000    0.872    0.958
##     VAA_LR            0.290    0.033    8.686    0.000    1.000    1.000
##     VAA_GT            0.167    0.026    6.398    0.000    1.000    1.000
##     CS_LR             0.068    0.025    2.666    0.008    1.000    1.000
##     CS_GT             0.127    0.029    4.436    0.000    1.000    1.000
##
## R-Square:
##                    Estimate
##     h26               0.309
##     h27               0.236
##     h25               0.306
##     h28               0.188
##     y19               0.171
##     h21               0.189
##     h22               0.249
##     h13               0.100
##     h29               0.152
##     h24               0.196
##     y25               0.259
```

```
##     C2b               0.077
##     C2g               0.285
##     C2h               0.637
##     C2a               0.210
##     C2c               0.155
##     C2d               0.153
##     C2e               0.024
##     C2f               0.124
##     C2i               0.299
##     C2j               0.042
##
## Defined Parameters:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     test.H3           0.076    0.025    3.028    0.002    0.647    0.647
##     test.H4           0.118    0.023    5.123    0.000    0.900    0.900
```

There is a Heywood correlation between GAL-TAN latent variables (absolute value > 1)

Respecified model: introduce the three preregistered residual correlations

```
model_H3H4.re<-paste0(model_H3H4,
                      "h27~~C2h\n",
                      "h21~~C2d\n",
                      "h29~~C2c\n")
```

Fitting the respecified model

```
fit_H3H4.re<-cfa(model=model_H3H4.re,
              data=dat2019.gmc,
              missing="fiml")
```

Results

Inspect fit of the model

```
round(inspect(fit_H3H4.re,"fit")
      [c("npar","df","chisq","pvalue","cfi","tli","rmsea","srmr")],3)
```

```
##    npar      df   chisq  pvalue     cfi     tli   rmsea    srmr
##  72.000 180.000 602.219   0.000   0.819   0.788   0.039   0.062
```

The fit of the model is quite poor according to CFI and TLI, but ok according to RMSEA and SRMR.

Hypotheses 1 and 2

Print standardized estimates to test the difference between correlations

```
std.est_H3H4.re<-standardizedsolution(fit_H3H4.re)
std.est_H3H4.re[std.est_H3H4.re$op==":=" |
               std.est_H3H4.re$op=="~~" &
               std.est_H3H4.re$lhs!=std.est_H3H4.re$rhs,]
```

```
##         lhs op                             rhs est.std    se      z pvalue
## 22   VAA_LR ~~                          VAA_GT   0.175 0.045  3.911  0.000
## 23    CS_LR ~~                           CS_GT   0.165 0.074  2.215  0.027
## 24   VAA_LR ~~                           CS_LR   0.782 0.055 14.220  0.000
## 25   VAA_GT ~~                           CS_GT   0.956 0.047 20.140  0.000
## 26   VAA_LR ~~                           CS_GT   0.189 0.068  2.772  0.006
## 27   VAA_GT ~~                           CS_LR   0.112 0.069  1.631  0.103
## 28      h27 ~~                             C2h   0.295 0.061  4.837  0.000
## 29      h21 ~~                             C2d   0.550 0.036 15.264  0.000
## 30      h29 ~~                             C2c   0.219 0.048  4.559  0.000
## 81 test.H3 := r.LR-max(r.VAA,r.CS,r.d1,r.d2)   0.593 0.086  6.863  0.000
## 82 test.H4 := r.GT-max(r.VAA,r.CS,r.d1,r.d2)   0.767 0.083  9.274  0.000
##    ci.lower ci.upper
## 22    0.087    0.262
## 23    0.019    0.310
## 24    0.674    0.890
## 25    0.863    1.049
## 26    0.055    0.323
## 27   -0.023    0.247
## 28    0.175    0.414
```

```
## 29    0.480    0.621
## 30    0.125    0.313
## 81    0.424    0.763
## 82    0.605    0.929
```

```r
#save to a file
write.csv2(std.est_H3H4.re[std.est_H3H4.re$op!="~1",c(1:5,7)],
           "std.est_H3H4.re.csv")
```

H3: There is strong (.782, $p < .001$) correlation between VAA-LR and CS-LR, and it is notably stronger (difference in correlations .593, $p < .001$) than the strongest of correlations between different dimensions (.189 between VAA_LR and VAA_GT, $p = .006$)

H4: There is very strong (.956, $p < .001$) correlation between VAA-GT and CS-GT, and it is notably stronger (difference in correlations .767, $p < .001$) than the strongest of correlations between different dimensions (.189 between VAA_LR and VAA_GT, $p = .006$)

Exploratory analysis for H3 and H4: Seek misspecification to improve the overall model fit

Residual correlations

```
mis.rescor_H3H4<-miPowerFit(fit_H3H4.re,cor=.20)
mis.rescor_H3H4<-mis.rescor_H3H4[mis.rescor_H3H4$op=="~~" &
                                    mis.rescor_H3H4$lhs!=mis.rescor_H3H4$rhs,]
#see summary of the decisions
table(mis.rescor_H3H4$decision.pow)
```

```
##
##  EPC:M EPC:NM      I     NM
##      2     43      1    161
```

```
#there are 2 residual correlation that are misspecifications

rounded.vars<-c("mi","epc","target.epc",
                "std.epc","se.epc")

num.round<-function(var){
  var<-as.numeric(var)
  var<-round(var,2)
  return(var)
}

mis.rescor_H3H4[,rounded.vars]<-sapply(mis.rescor_H3H4[,rounded.vars],num.round)

printed.vars<-c("lhs","op","rhs","mi","epc","target.epc",
                "std.epc","std.target.epc","significant.mi",
                "high.power","decision.pow","se.epc")

#print the output

mis.rescor_H3H4 %>%
  filter(mis.rescor_H3H4$decision.pow=="M" |
                mis.rescor_H3H4$decision.pow=="EPC:M") %>%
  dplyr::select(all_of(printed.vars))
```

```
##   lhs op rhs    mi  epc target.epc std.epc std.target.epc significant.mi
## 1 h25 ~~ y19 91.85 0.17       0.13    0.26            0.2           TRUE
## 2 C2a ~~ C2f 26.23 0.17       0.16    0.22            0.2           TRUE
##   high.power decision.pow se.epc
## 1       TRUE        EPC:M   0.02
## 2       TRUE        EPC:M   0.03
```

There were two misspecified residual correlation.

One was between VAA-LR items (same misspecification as was found for H1 and H2) H25. Public services should be outsourced more than they are now for private companies and y19. Public authorities should be the main provider of social and healthcare services (r.)

The other misspecification was between C2a. Immigrants should adapt to Finnish habits and C2f. People who break the law should be punished more severely

Respecify the model to allow these parameters to be freely estimated

Exploratory respecification of the model

```
model_H3H4.exp.re<-paste0(model_H3H4.re,
                         "h25~~y19\n",
                         "C2a~~C2f\n")
```

Fitting the exploratory model

```
fit_H3H4.exp.re<-cfa(model=model_H3H4.exp.re,
              data=dat2019.gmc,
              missing="fiml")
```

Results from the exploratory model

```
round(inspect(fit_H3H4.re,"fit")
      [c("npar","df","chisq","pvalue","cfi","tli","rmsea","srmr")],3)
```

```
##    npar      df   chisq  pvalue     cfi     tli   rmsea    srmr
##  72.000 180.000 602.219   0.000   0.819   0.788   0.039   0.062
```

```
round(inspect(fit_H3H4.exp.re,"fit")
      [c("npar","df","chisq","pvalue","cfi","tli","rmsea","srmr")],3)
```

```
##    npar      df   chisq  pvalue     cfi     tli   rmsea    srmr
##  74.000 178.000 488.872   0.000   0.866   0.842   0.033   0.059
```

The fit of the model is improved

Retest Hypotheses 4 and 5

Print standardized estimates to test the difference between correlations

```
std.est_H3H4.exp.re<-standardizedsolution(fit_H3H4.exp.re)
std.est_H3H4.exp.re[std.est_H3H4.exp.re$op==":=" |
               std.est_H3H4.exp.re$op=="~~" &
               std.est_H3H4.exp.re$lhs!=std.est_H3H4.exp.re$rhs,]
```

```
##        lhs op                           rhs est.std    se      z pvalue
## 22  VAA_LR ~~                        VAA_GT   0.239 0.045  5.367  0.000
## 23   CS_LR ~~                         CS_GT   0.212 0.075  2.835  0.005
## 24  VAA_LR ~~                         CS_LR   0.831 0.055 15.003  0.000
## 25  VAA_GT ~~                         CS_GT   0.957 0.052 18.422  0.000
## 26  VAA_LR ~~                         CS_GT   0.220 0.072  3.074  0.002
## 27  VAA_GT ~~                         CS_LR   0.127 0.068  1.875  0.061
## 28     h27 ~~                           C2h   0.246 0.065  3.819  0.000
## 29     h21 ~~                           C2d   0.546 0.037 14.956  0.000
## 30     h29 ~~                           C2c   0.215 0.049  4.434  0.000
## 31     h25 ~~                           y19   0.290 0.028 10.525  0.000
## 32     C2a ~~                           C2f   0.253 0.047  5.439  0.000
## 83 test.H3 := r.LR-max(r.VAA,r.CS,r.d1,r.d2)   0.592 0.071  8.324  0.000
## 84 test.H4 := r.GT-max(r.VAA,r.CS,r.d1,r.d2)   0.718 0.067 10.651  0.000
##    ci.lower ci.upper
## 22    0.152    0.327
## 23    0.065    0.359
```

```
## 24    0.723    0.940
## 25    0.856    1.059
## 26    0.080    0.360
## 27   -0.006    0.260
## 28    0.120    0.373
## 29    0.475    0.618
## 30    0.120    0.311
## 31    0.236    0.344
## 32    0.162    0.345
## 83    0.452    0.731
## 84    0.586    0.850
```

```
#save to a file
write.csv2(std.est_H3H4.exp.re[std.est_H3H4.exp.re$op!="~1",c(1:5,7)],
           "std.est_H3H4.exp.re.csv")
```

The results are virtually identical to those without the additional residual correlations.

H3: There is a strong (.831, p < .001) correlation between VAA-LR and CS-LR, and it is notably stronger (difference in correlations .592, p < .001) than the strongest of correlations between different dimensions (.239 between VAA_LR and VAA_GT, p < .001)

H4: There is a very strong (.957, p < .001) correlation between VAA-GT and CS-GT, and it is notably stronger (difference in correlations .718, p < .001) than the strongest of correlations between different dimensions (.239 between VAA_LR and VAA_GT, p < .001)

Additional explorations through more strict respecifications

Put a more strict criterion on the residual correlation misspecification (.15)

```
mis.rescor_H3H4<-miPowerFit(fit_H3H4.exp.re,cor=.15)
mis.rescor_H3H4<-mis.rescor_H3H4[mis.rescor_H3H4$op=="~~" &
                                   mis.rescor_H3H4$lhs!=mis.rescor_H3H4$rhs,]
#see summary of the decisions
table(mis.rescor_H3H4$decision.pow)
```

```
##
##  EPC:M EPC:NM      I     NM
##      2     34      1    168
```

```
#there are two additional residual correlations that are misspecifications

rounded.vars<-c("mi","epc","target.epc",
                "std.epc","se.epc")

num.round<-function(var){
  var<-as.numeric(var)
  var<-round(var,2)
  return(var)
}

mis.rescor_H3H4[,rounded.vars]<-sapply(mis.rescor_H3H4[,rounded.vars],num.round)

printed.vars<-c("lhs","op","rhs","mi","epc","target.epc",
                "std.epc","std.target.epc","significant.mi",
                "high.power","decision.pow","se.epc")
```

```r
#print the output

mis.rescor_H3H4 %>%
  filter(mis.rescor_H3H4$decision.pow=="M" |
                mis.rescor_H3H4$decision.pow=="EPC:M") %>%
  dplyr::select(all_of(printed.vars))
```

```
##   lhs op rhs    mi  epc target.epc std.epc std.target.epc significant.mi
## 1 h22 ~~ C2i 16.44 0.16       0.14    0.17           0.15           TRUE
## 2 C2d ~~ C2j 24.41 0.17       0.14    0.18           0.15           TRUE
##   high.power decision.pow se.epc
## 1       TRUE        EPC:M   0.04
## 2       TRUE        EPC:M   0.03
```

There were two more misspecified residual correlations.

Between VAA-GAL-TAN h22. If the government proposes to establish a refugee center in my home municipality, the proposal should be accepted (r.) and CS-GAL-TAN C2i. Immigrants are good for the Finnish economy (r.)

And between two CS-GAL-TAN items: C2d. Same Sex Marriages should be prohibited by law and C2j. Deciding on abortion issues should be a women's right (r.)

Respecify the model 2

```r
model_H3H4.exp.re.2<-paste0(model_H3H4.exp.re,
                      "h22~~C2i\n",
                      "C2d~~C2j\n")
```

Fit the respecified model 2

```r
fit_H3H4.exp.re.2<-cfa(model=model_H3H4.exp.re.2,
              data=dat2019.gmc,
              missing="fiml")
```

Results from the exploratory model 2

```r
round(inspect(fit_H3H4.exp.re,"fit")
      [c("npar","df","chisq","pvalue","cfi","tli","rmsea","srmr")],3)
```

```
##    npar      df   chisq  pvalue     cfi     tli   rmsea    srmr
##  74.000 178.000 488.872   0.000   0.866   0.842   0.033   0.059
```

```r
round(inspect(fit_H3H4.exp.re.2,"fit")
      [c("npar","df","chisq","pvalue","cfi","tli","rmsea","srmr")],3)
```

```
##    npar      df   chisq  pvalue     cfi     tli   rmsea    srmr
##  76.000 176.000 446.837   0.000   0.884   0.861   0.031   0.056
```

The fit of the model is again improved

Retest Hypotheses 4 and 5

Print standardized estimates to test the difference between correlations

```r
std.est_H3H4.exp<-standardizedsolution(fit_H3H4.exp.re.2)
std.est_H3H4.exp[std.est_H3H4.exp$op==":=" |
               std.est_H3H4.exp$op=="~~" &
               std.est_H3H4.exp$lhs!=std.est_H3H4.exp$rhs,]
```

```
##        lhs op                              rhs est.std    se      z pvalue
## 22  VAA_LR ~~                           VAA_GT   0.236 0.045  5.274  0.000
## 23   CS_LR ~~                            CS_GT   0.183 0.079  2.335  0.020
## 24  VAA_LR ~~                            CS_LR   0.830 0.056 14.939  0.000
## 25  VAA_GT ~~                            CS_GT   0.945 0.053 17.802  0.000
## 26  VAA_LR ~~                            CS_GT   0.197 0.074  2.659  0.008
## 27  VAA_GT ~~                            CS_LR   0.121 0.068  1.780  0.075
## 28     h27 ~~                              C2h   0.246 0.065  3.797  0.000
## 29     h21 ~~                              C2d   0.515 0.039 13.210  0.000
## 30     h29 ~~                              C2c   0.214 0.049  4.376  0.000
## 31     h25 ~~                              y19   0.290 0.028 10.522  0.000
## 32     C2a ~~                              C2f   0.243 0.049  5.013  0.000
## 33     h22 ~~                              C2i   0.233 0.054  4.335  0.000
## 34     C2d ~~                              C2j   0.205 0.041  5.013  0.000
## 85 test.H3 := r.LR-max(r.VAA,r.CS,r.d1,r.d2)   0.594 0.071  8.332  0.000
## 86 test.H4 := r.GT-max(r.VAA,r.CS,r.d1,r.d2)   0.709 0.069 10.330  0.000
##    ci.lower ci.upper
## 22    0.148    0.324
## 23    0.029    0.337
## 24    0.721    0.939
## 25    0.841    1.049
## 26    0.052    0.342
## 27   -0.012    0.255
## 28    0.119    0.373
## 29    0.439    0.592
## 30    0.118    0.309
## 31    0.236    0.344
## 32    0.148    0.338
## 33    0.128    0.338
## 34    0.125    0.285
## 85    0.455    0.734
## 86    0.574    0.843
```

Nothing important changed in the latent correlations

H6 with group mean centered observed variables

H6. Within-party placement on Left-Right as computed from responses to the pre-election public Voting Advice Applications (VAAs) is positively associated with within-party placement on Left-Right as computed from responses to the privately administered post-election Candidate Survey (CS). This association is stronger than any within-party associations between the Left-Right and GAL-TAN dimensions.

Add placement variables and their correlations with latent factors to the model used for H3 and H4

Model already includes the three preregistered correlations

```
model_H6<-paste0(model_H3H4.re,
                 "SP_LR=~C5a\n",
                 "IP_LR=~C5c\n",
                 "VAA_LR~~r.self.LR*SP_LR\n",
                 "VAA_LR~~r.ideal.LR*IP_LR\n",
                 "test.H6:=r.self.LR-r.ideal.LR\n")
```

Fit the model

```
fit_H6<-cfa(model=model_H6,
            data=dat2019.gmc,
            missing="fiml")
```

Inspect fit of the model

```
round(inspect(fit_H6,"fit")
      [c("npar","df","chisq","pvalue","cfi","tli","rmsea","srmr")],3)
```

```
##    npar      df   chisq  pvalue     cfi     tli   rmsea    srmr
##  85.000 214.000 679.220   0.000   0.818   0.785   0.037   0.061
```

The fit of the model is ok based on rmsea and srmr, but poor according to cfi and tli

Hypothesis 6

Print standardized estimates to test the difference between correlations

```
std.est_H6<-standardizedsolution(fit_H6)
std.est_H6[std.est_H6$op==":=" |
               std.est_H6$op=="~~" &
               std.est_H6$lhs!=std.est_H6$rhs,]
```

```
##         lhs op                            rhs est.std    se      z pvalue
## 22   VAA_LR ~~                         VAA_GT   0.176 0.045  3.947  0.000
## 23    CS_LR ~~                          CS_GT   0.171 0.075  2.279  0.023
## 24   VAA_LR ~~                          CS_LR   0.796 0.053 15.109  0.000
## 25   VAA_GT ~~                          CS_GT   0.955 0.047 20.128  0.000
## 26   VAA_LR ~~                          CS_GT   0.187 0.068  2.742  0.006
## 27   VAA_GT ~~                          CS_LR   0.119 0.069  1.715  0.086
## 28      h27 ~~                            C2h   0.287 0.057  4.993  0.000
## 29      h21 ~~                            C2d   0.549 0.036 15.203  0.000
## 30      h29 ~~                            C2c   0.220 0.048  4.581  0.000
## 33   VAA_LR ~~                          SP_LR   0.469 0.050  9.321  0.000
## 34   VAA_LR ~~                          IP_LR   0.069 0.060  1.150  0.250
```

```
## 64   VAA_GT ~~                          SP_LR   0.217 0.057  3.776  0.000
## 65   VAA_GT ~~                          IP_LR   0.084 0.062  1.354  0.176
## 66    CS_LR ~~                          SP_LR   0.446 0.051  8.830  0.000
## 67    CS_LR ~~                          IP_LR   0.124 0.057  2.194  0.028
## 68    CS_GT ~~                          SP_LR   0.188 0.058  3.210  0.001
## 69    CS_GT ~~                          IP_LR   0.090 0.061  1.478  0.139
## 70    SP_LR ~~                          IP_LR   0.423 0.038 11.199  0.000
## 100 test.H3 := r.LR-max(r.VAA,r.CS,r.d1,r.d2)   0.609 0.085  7.165  0.000
## 101 test.H4 := r.GT-max(r.VAA,r.CS,r.d1,r.d2)   0.768 0.083  9.289  0.000
## 102 test.H6 :=          r.self.LR-r.ideal.LR   0.400 0.060  6.660  0.000
##     ci.lower ci.upper
## 22     0.089    0.263
## 23     0.024    0.319
## 24     0.693    0.899
## 25     0.862    1.048
## 26     0.053    0.321
## 27    -0.017    0.255
## 28     0.174    0.399
## 29     0.478    0.620
## 30     0.126    0.314
## 33     0.370    0.568
## 34    -0.049    0.186
## 64     0.104    0.329
## 65    -0.038    0.207
## 66     0.347    0.546
## 67     0.013    0.235
## 68     0.073    0.302
## 69    -0.029    0.210
## 70     0.349    0.497
## 100    0.442    0.775
## 101    0.606    0.930
## 102    0.282    0.518
```

```
#save to a file
write.csv2(std.est_H6[std.est_H6$op!="~1",c(1:5,7)],
           "std.est_H6.csv")
```

H6. The correlation between VAA_LR and CS Self-placement on LR is strong (.469, $p < .001$) and stronger than the association between VAA_LR and placement of imagined party voter (.069, $p = .250$; difference .400, $p < .001$)

Exploratory analysis of H6: Look for misspecifications

Residual correlations

```
mis.rescor_H6<-miPowerFit(fit_H6,cor=.20)
mis.rescor_H6<-mis.rescor_H6[mis.rescor_H6$op=="~~" &
                                    mis.rescor_H6$lhs!=mis.rescor_H6$rhs,]
#see summary of the decisions
table(mis.rescor_H6$decision.pow)
```

```
##
##   EPC:M EPC:NM      I     NM
##       2     49      1    197
```

```r
#there are is a single misspecification with .15 as criterion

rounded.vars<-c("mi","epc","target.epc",
                "std.epc","se.epc")

num.round<-function(var){
  var<-as.numeric(var)
  var<-round(var,2)
  return(var)
}

mis.rescor_H6[,rounded.vars]<-sapply(mis.rescor_H6[,rounded.vars],num.round)

printed.vars<-c("lhs","op","rhs","mi","epc","target.epc",
                "std.epc","std.target.epc","significant.mi",
                "high.power","decision.pow","se.epc")

#print the output

mis.rescor_H6 %>%
  filter(mis.rescor_H6$decision.pow=="M" |
               mis.rescor_H6$decision.pow=="EPC:M") %>%
  dplyr::select(all_of(printed.vars))
```

```
##   lhs op rhs    mi  epc target.epc std.epc std.target.epc significant.mi
## 1 h25 ~~ y19 92.01 0.17       0.13    0.26            0.2           TRUE
## 2 C2a ~~ C2f 25.51 0.17       0.16    0.21            0.2           TRUE
##   high.power decision.pow se.epc
## 1       TRUE        EPC:M   0.02
## 2       TRUE        EPC:M   0.03
```

Same misspecifications as there were for model for H3 and H4

Add to the model

```r
model_H6.re<-paste0(model_H6,
                      "h25~~y19\n",
                    "C2a~~C2f")
```

Fit the respecified model

```r
fit_H6.re<-cfa(model=model_H6.re,
              data=dat2019.gmc,
              missing="fiml")
```

Results

```r
round(inspect(fit_H6,"fit")
      [c("npar","df","chisq","pvalue","cfi","tli","rmsea","srmr")],3)
```

```
##    npar      df   chisq  pvalue     cfi     tli   rmsea    srmr
##  85.000 214.000 679.220   0.000   0.818   0.785   0.037   0.061
```

```r
round(inspect(fit_H6.re,"fit")
      [c("npar","df","chisq","pvalue","cfi","tli","rmsea","srmr")],3)
```

```
##    npar      df   chisq  pvalue     cfi     tli   rmsea    srmr
##  87.000 212.000 565.706   0.000   0.862   0.835   0.033   0.058
```

Fit is improved.

Print standardized estimates to test the difference between correlations

```r
std.est_H6.re<-standardizedsolution(fit_H6.re)
std.est_H6.re[std.est_H6.re$op==":=" |
               std.est_H6.re$op=="~~" &
               std.est_H6.re$lhs!=std.est_H6.re$rhs,]
```

```
##         lhs op                            rhs est.std    se      z pvalue
## 22   VAA_LR ~~                         VAA_GT   0.239 0.045  5.349  0.000
## 23    CS_LR ~~                          CS_GT   0.222 0.076  2.940  0.003
## 24   VAA_LR ~~                          CS_LR   0.842 0.053 15.767  0.000
## 25   VAA_GT ~~                          CS_GT   0.958 0.052 18.423  0.000
## 26   VAA_LR ~~                          CS_GT   0.219 0.072  3.058  0.002
## 27   VAA_GT ~~                          CS_LR   0.133 0.069  1.936  0.053
## 28      h27 ~~                            C2h   0.250 0.060  4.152  0.000
## 29      h21 ~~                            C2d   0.544 0.037 14.854  0.000
## 30      h29 ~~                            C2c   0.216 0.049  4.438  0.000
## 33   VAA_LR ~~                          SP_LR   0.494 0.051  9.605  0.000
## 34   VAA_LR ~~                          IP_LR   0.072 0.062  1.158  0.247
## 35      h25 ~~                            y19   0.290 0.028 10.523  0.000
## 36      C2a ~~                            C2f   0.252 0.047  5.375  0.000
## 66   VAA_GT ~~                          SP_LR   0.216 0.057  3.770  0.000
## 67   VAA_GT ~~                          IP_LR   0.079 0.062  1.269  0.204
## 68    CS_LR ~~                          SP_LR   0.447 0.050  8.891  0.000
## 69    CS_LR ~~                          IP_LR   0.124 0.056  2.204  0.028
## 70    CS_GT ~~                          SP_LR   0.183 0.060  3.044  0.002
## 71    CS_GT ~~                          IP_LR   0.081 0.063  1.294  0.196
## 72    SP_LR ~~                          IP_LR   0.423 0.038 11.199  0.000
## 102 test.H3 := r.LR-max(r.VAA,r.CS,r.d1,r.d2)   0.604 0.070  8.670  0.000
## 103 test.H4 := r.GT-max(r.VAA,r.CS,r.d1,r.d2)   0.720 0.067 10.664  0.000
## 104 test.H6 :=           r.self.LR-r.ideal.LR   0.422 0.062  6.846  0.000
##     ci.lower ci.upper
## 22     0.151    0.326
## 23     0.074    0.370
## 24     0.738    0.947
## 25     0.856    1.060
## 26     0.079    0.360
## 27    -0.002    0.267
## 28     0.132    0.367
## 29     0.473    0.616
## 30     0.120    0.311
## 33     0.393    0.595
## 34    -0.049    0.193
## 35     0.236    0.344
## 36     0.160    0.344
## 66     0.104    0.329
## 67    -0.043    0.202
```

```
## 68      0.348    0.545
## 69      0.014    0.235
## 70      0.065    0.301
## 71     -0.042    0.204
## 72      0.349    0.497
## 102     0.467    0.740
## 103     0.587    0.852
## 104     0.302    0.543
```

```r
#save to a file
write.csv2(std.est_H6.re[std.est_H6.re$op!="~1",c(1:5,7)],
           "std.est_H6.re.csv")
```

Results are virtually identical.

H6. The correlation between VAA_LR and CS Self-placement on LR is moderately strong (.494, $p < .001$) and stronger than the association between VAA_LR and placement of imagined party voter (.072, $p = .247$; difference .422, $p < .001$)

Some comparison between overall and unconfounded models

Extract some parameters from the confirmatory models for H1-H2 and H3-H4

```
load.overall<-std.est_H1H2[std.est_H1H2$op=="=~",1:4]
load.unconfounded<-std.est_H3H4.re[std.est_H3H4.re$op=="=~",1:4]

describe(load.overall[,4]-load.unconfounded[,4])
```

```
##    vars  n mean   sd median trimmed  mad  min  max range skew kurtosis   se
## X1    1 21 0.21 0.08    0.2     0.2 0.08 0.07 0.33  0.26 0.11    -1.32 0.02
```

```
corr.test(load.overall[,4],load.unconfounded[,4],adjust="none")
```

```
## Call:corr.test(x = load.overall[, 4], y = load.unconfounded[, 4],
##     adjust = "none")
## Correlation matrix
## [1] 0.85
## Sample Size
## [1] 21
## [1] 0
##
##  To see confidence intervals of the correlations, print with the short=FALSE option
```

```
lv.cor.cross.overall<-std.est_H1H2[std.est_H1H2$op=="~~" &
                             std.est_H1H2$lhs!=std.est_H1H2$rhs &
                             ((grepl("LR",std.est_H1H2$lhs) &
                                 grepl("GT",std.est_H1H2$rhs))|
                                (grepl("GT",std.est_H1H2$lhs) &
                                   grepl("LR",std.est_H1H2$rhs))),1:4]

lv.cor.cross.unconfounded<-std.est_H3H4.re[std.est_H3H4.re$op=="~~" &
                             std.est_H3H4.re$lhs!=std.est_H3H4.re$rhs &
                             ((grepl("LR",std.est_H3H4.re$lhs) &
                                 grepl("GT",std.est_H3H4.re$rhs))|
                                (grepl("GT",std.est_H3H4.re$lhs) &
                                   grepl("LR",std.est_H3H4.re$rhs))),1:4]
mean(lv.cor.cross.overall[,4])
```

```
## [1] 0.3810004
```

```
mean(lv.cor.cross.unconfounded[,4])
```

```
## [1] 0.1601159
```

```
describe(lv.cor.cross.overall[,4]-lv.cor.cross.unconfounded[,4])
```

```
##    vars n mean   sd median trimmed  mad  min  max range  skew kurtosis   se
## X1    1 4 0.22 0.02   0.22    0.22 0.02 0.19 0.25  0.06 -0.12    -1.92 0.01
```

Figures

Obtain factor scores and plot the against each other for VAA and CS

```
#CS items must be converted to numeric from haven_labelled to calculate factor scores
pred.dat2019<-dat2019 %>%
  mutate_at(c(CS_GT_items,
              CS_LR_items),as.numeric)

#refit the model with converted data
fit_H1H2.re.pred<-cfa(model=model_H1H2.re,
                      data=pred.dat2019,
                      missing="fiml")

#only complete cases from the eight parties are used for the plot
plot.dat2019<-pred.dat2019 %>%
  select(all_of(VAA_LR_items),
         all_of(VAA_GT_items),
         all_of(CS_LR_items),
         all_of(CS_GT_items),
         puolue) %>%
  filter(puolue=="KD" |
           puolue=="KESK" |
           puolue=="KOK" |
           puolue=="PS" |
           puolue=="RKP" |
           puolue=="SDP" |
           puolue=="VAS" |
           puolue=="VIHR") %>%
  na.omit() %>%
  mutate(puolue=as.character(puolue))

#calculate factor scores
scores<-
  data.frame(lavPredict(fit_H1H2.re.pred,
                        newdata=plot.dat2019,
                        type="lv",append.data=T,
                        method="Bartlett"),
             puolue=plot.dat2019$puolue)

#standardize scores

scores[,1:4]<-scale(scores[,1:4],center=T,scale=T)
```

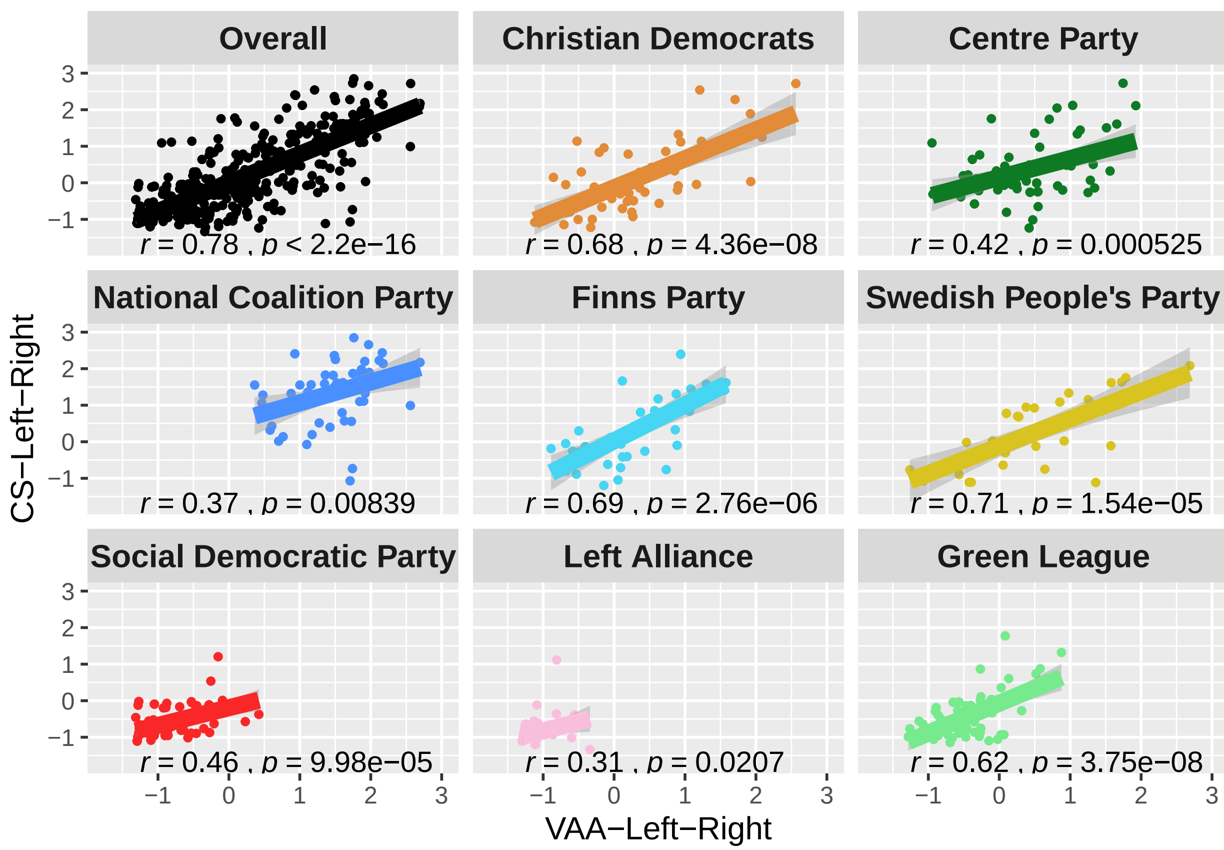
```r
#set up colors
puolue.varit.8<-c(KD = "#e28b38",KESK = "#0e7a24",KOK = "#498fff",PS = "#46d5f2",
                  RKP = "#d8c320",SDP = "#f92727",VAS = "#fabedd",VIHR = "#77ea8e")
#rename puolue to party
scores$Party<-scores$puolue
#construct a long dataframe with different observations for overall data
scores.2<-scores
scores.2$Party<-"ALL"
scores.long<-rbind(scores.2,scores)

scores.long$Party<-as.factor(scores.long$Party)
levels(scores.long$Party)<-c("Overall",
                             "Christian Democrats",
                             "Centre Party",
                             "National Coalition Party",
                             "Finns Party",
                             "Swedish People's Party",
                             "Social Democratic Party",
                             "Left Alliance",
                             "Green League")
```

LR Plot

```
LR.plot<-ggplot(data=scores.long,aes(x=VAA_LR,y=CS_LR,color=Party))+
  geom_point(size=1)+
  geom_smooth(formula = y ~ x,method="lm",size=3,linetype=1)+
  stat_cor(method = "pearson", label.x = -1.25, label.y = -1.75,color="black",
           digits = 2,
           r.digits = 2,
           p.digits = 3,
           cor.coef.name="r")+
  scale_color_manual(values=c("black",puolue.varit.8))+
  coord_cartesian(xlim=c(-1.75,3),ylim=c(-1.75,3))+
  xlab("VAA-Left-Right")+
  ylab("CS-Left-Right")+
  theme(legend.position = "none",axis.title = element_text(size=12))+
  facet_wrap(~Party)+
  theme(strip.text.x =
          element_text(size = 12, face = "bold"))

LR.plot
```



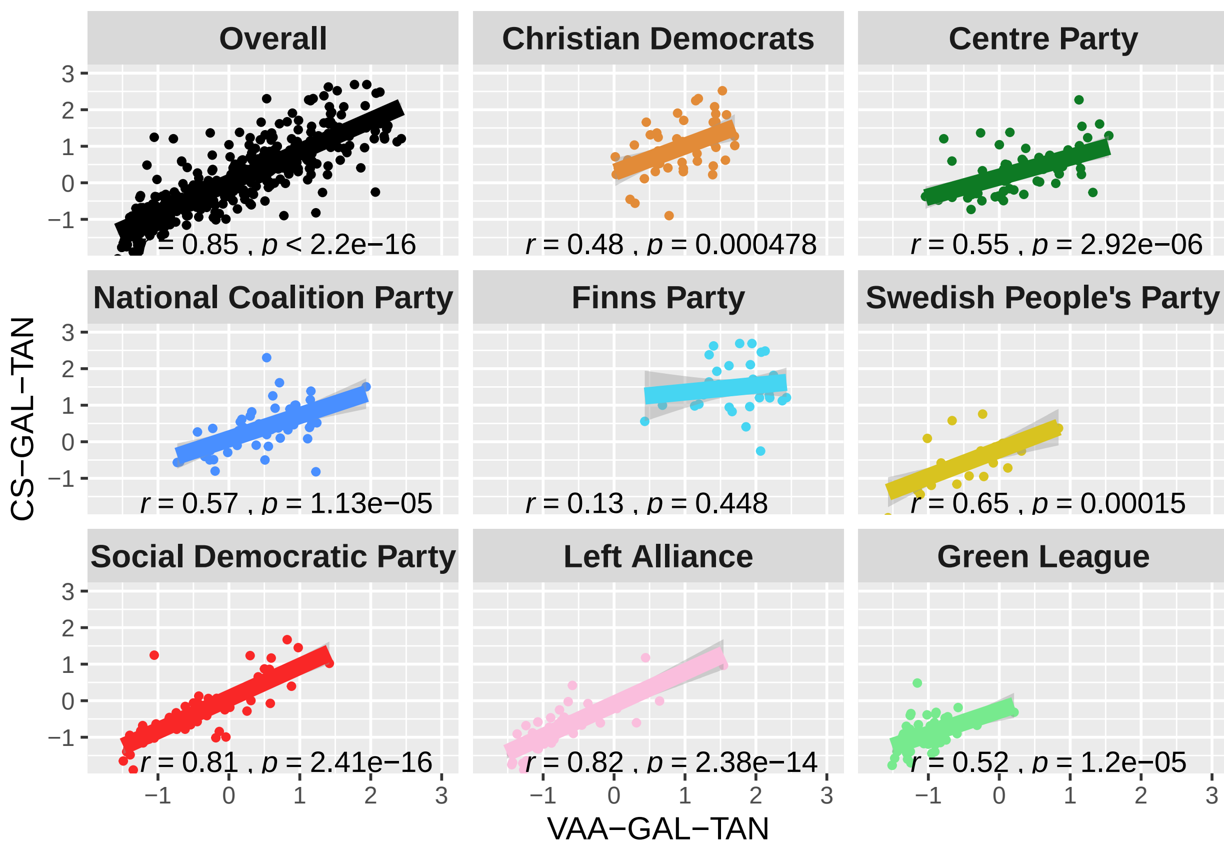

```
#save the plot
ggsave(plot=LR.plot,
       filename="LR.plot.png",
       width=24,
       height=24,
```

```
       units = "cm",
       device = "png")
```

GT plot

```
GT.plot<-ggplot(data=scores.long,aes(x=VAA_GT,y=CS_GT,color=Party))+
  geom_point(size=1)+
  geom_smooth(formula = y ~ x,method="lm",size=3,linetype=1)+
  stat_cor(method = "pearson", label.x = -1.25, label.y = -1.75,color="black",
           digits = 2,
           r.digits = 2,
           p.digits = 3,
           cor.coef.name="r")+
  scale_color_manual(values=c("black",puolue.varit.8))+
  coord_cartesian(xlim=c(-1.75,3),ylim=c(-1.75,3))+
  xlab("VAA-GAL-TAN")+
  ylab("CS-GAL-TAN")+
  theme(legend.position = "none",axis.title = element_text(size=12))+
  facet_wrap(~Party)+
  theme(strip.text.x =
          element_text(size = 12, face = "bold"))

GT.plot
```




```
ggsave(plot=GT.plot,
       filename="GT.plot.png",
       width=24,
       height=24,
       units = "cm",
```

```
       device = "png")
```

Session information

```r
sessionInfo()
```

```
## R version 3.6.3 (2020-02-29)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 17763)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=Finnish_Finland.1252  LC_CTYPE=Finnish_Finland.1252
## [3] LC_MONETARY=Finnish_Finland.1252 LC_NUMERIC=C
## [5] LC_TIME=Finnish_Finland.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] ggpubr_0.3.0     ggrepel_0.8.2    sjlabelled_1.1.3 haven_2.2.0
##  [5] semPlot_1.1.2    semTools_0.5-2   lavaan_0.6-5     psych_1.9.12.31
##  [9] stringr_1.4.0    tidyr_1.1.0      ggplot2_3.3.0    labelled_2.2.2
## [13] dplyr_0.8.5      here_0.1
##
## loaded via a namespace (and not attached):
##   [1] minqa_1.2.4         colorspace_1.4-1    ggsignif_0.6.0
##   [4] rjson_0.2.20        rio_0.5.16          ellipsis_0.3.0
##   [7] rprojroot_1.3-2     htmlTable_1.13.3    corpcor_1.6.9
##  [10] base64enc_0.1-3     rstudioapi_0.11     farver_2.0.3
##  [13] splines_3.6.3       mnormt_1.5-6        knitr_1.28
##  [16] glasso_1.11         Formula_1.2-3       nloptr_1.2.2.1
##  [19] broom_0.5.6         cluster_2.1.0       png_0.1-7
##  [22] regsem_1.5.2        compiler_3.6.3      backports_1.1.6
##  [25] assertthat_0.2.1    Matrix_1.2-18       acepack_1.4.1
##  [28] htmltools_0.4.0     tools_3.6.3         igraph_1.2.5
##  [31] OpenMx_2.17.3       coda_0.19-3         gtable_0.3.0
##  [34] glue_1.4.1          reshape2_1.4.4      Rcpp_1.0.4.6
##  [37] carData_3.0-3       cellranger_1.1.0    vctrs_0.3.0
##  [40] nlme_3.1-144        lisrelToR_0.1.4     multilevel_2.6
##  [43] insight_0.8.3       xfun_0.13           openxlsx_4.1.4
##  [46] lme4_1.1-23         lifecycle_0.2.0     gtools_3.8.2
##  [49] rstatix_0.5.0       statmod_1.4.34      XML_3.99-0.3
##  [52] MASS_7.3-51.5       scales_1.1.1        BDgraph_2.62
##  [55] hms_0.5.3           kutils_1.69         parallel_3.6.3
##  [58] huge_1.3.4.1        RColorBrewer_1.1-2  curl_4.3
##  [61] yaml_2.2.1          pbapply_1.4-2       gridExtra_2.3
##  [64] rpart_4.1-15        latticeExtra_0.6-29 stringi_1.4.6
##  [67] sem_3.1-9           checkmate_2.0.0     boot_1.3-24
##  [70] zip_2.0.4           truncnorm_1.0-8     rlang_0.4.6
##  [73] pkgconfig_2.0.3     d3Network_0.5.2.1   Rsolnp_1.16
##  [76] arm_1.10-1          evaluate_0.14       lattice_0.20-38
##  [79] purrr_0.3.4         labeling_0.3        htmlwidgets_1.5.1
##  [82] tidyselect_1.1.0    plyr_1.8.6          magrittr_1.5
```

```
##  [85] R6_2.4.1            generics_0.0.2      Hmisc_4.4-0
##  [88] mgcv_1.8-31         pillar_1.4.3        whisker_0.4
##  [91] foreign_0.8-75      withr_2.2.0         rockchalk_1.8.144
##  [94] survival_3.1-8      abind_1.4-5         nnet_7.3-12
##  [97] tibble_3.0.1        car_3.0-7           crayon_1.3.4
## [100] fdrtool_1.2.15      rmarkdown_2.1       jpeg_0.1-8.1
## [103] readxl_1.3.1        grid_3.6.3          qgraph_1.6.5
## [106] data.table_1.12.8   pbivnorm_0.6.0      forcats_0.5.0
## [109] matrixcalc_1.0-3    digest_0.6.25       xtable_1.8-4
## [112] mi_1.0              stats4_3.6.3        munsell_0.5.0
```