COLA Registered Report: Sample size justification

4th September 2020

# Registered report: Inconsistent language lateralisation: evidence from behaviour and lateralised cerebral blood flow

## The COLA Consortium

## Hypotheses

Hypotheses

The overarching hypothesis is that there are cross-hemispheric dissociations in lateralisation of different language functions, especially in left handers. A subsidiary hypothesis, derived from Woodhead et al (2019, 2020), is that laterality indices from tasks that primarily implicate language generation and receptive language will form separate clusters.

### Predictions

*Online behavioural measures*  
1. On online language tasks that show left-lateralisation for language, lateralisation will be stronger in right than left handers, in terms of central tendency as well as the proportion of participants who show the typical side bias (cf. Karlsson, Johnstone & Carey, 2019).

1. The pattern of correlation between laterality indices from online measures will reflect the extent to which they involve implicit speech production, rather than whether they involve spoken or written language. Thus we anticipate dissociation between the rhyme decision task, which involves implicit speech generation, and the word comprehension task and dichotic listening tasks, which do not. We further anticipate that dissociations between tasks are not accountable for in terms of low reliability of measures – i.e. correlations of laterality indices between tasks will be lower than split-half reliability of the measures.

*FTCD measures*  
3. The data will fit a model where ‘language generation’ tasks cluster together on one factor, and ‘receptive’ language tasks on a second factor. The factors will be correlated, but the fit of a 2-factor model will be superior to a single-factor model.

1. Following Woodhead et al (2020), we predict that better model fit will be obtained when different parameters are estimated for left- vs right handers, compared with when all parameters are held constant.
2. On categorical analysis, individuals who depart from left-brained laterality on one or more tasks will be more likely to be left-handed than those who are consistently left-lateralised.

*Relationship between fTCD and behavioural laterality indices*  
6. The laterality profile obtained with the online language battery will be significantly associated with the profile seen with direct measure of cerebral blood flow using fTCD, with laterality on dichotic listening relating more strongly to receptive language tasks, and visual-half-field language measures to language generation tasks.

# Sample size justification

The sample size is determined not solely by statistical power, though we aim for 90% power for key analyses. In addition, we take into consideration the fact that this is a two-stage study, with online testing followed by in-person testing. Recruitment for online testing is considerably easier and less costly than in-person testing. The timing of in-person testing remains uncertain due to pandemic restrictions, and this means that we may see considerable attrition between stages 1 and 2. Our strategy is first to determine sample size based on power considerations for stage 2, then to double that sample size for online testing, assuming there will be 50% attrition between stages 1 and 2. In addition, we check the power that this gives us for stage 1 testing. Accordingly, we report power analyses in reverse order for stage 2 and then stage 1.

## Single group confirmatory factor analysis models (one vs two factor)

Sample size determination has two aspects: 1. We need to have sufficient sample size to fit individual models with acceptable fit indices, regardless of whether these are one factor or two factor; 2. There should be adequate power to detect the misspecified model (one factor) vs the ‘true’ model (two factor). For the first point, we use the Monte Carlo simulation results for Confirmatory Factor Analysis (CFA) presented by Wolf et al (2013) @Wolf\_2013. These authors reported results from a range of simple CFA models, varying numbers of factors, numbers of measured variables, and magnitudes of factor loadings, in order to evaluate statistical power, bias in parameter estimates and incidence of model convergence problems. Our proposed model comes closest to their simulation of a 2-factor model with 3 indicators per factor; the minimum sample size required to achieve 80% power depends on the strength of factor loadings, ranging between 120-200 for loadings of .65 or .8 respectively. This is in line with our next analysis, which estimates power to detect the difference between a one- and two-factor model.

### Power estimate for the model comparison of proposed ‘true’ model vs proposed mispecified model.

The R package semPower [@Moshagen\_2016] can be used to calculate the power for detecting a difference between nested models, one factor vs two factors, using data simulated from both models. Our model comparison is closely similar to that used in the vignette for the semPower package by @Moshagen\_2020. Moshagen (2020) provides one example where the interest is in whether responses on 8 items reflect two separate but correlated factors, or whether they can be described by a single factor; this is directly parallel to our prediction 3.

Moshagen (2020) provides a further example that illustrates a power test for invariance constraints in a multiple group model, where the question is whether a model with freely estimated loadings performs better than a model where loadings are equal for two groups: this addresses our prediction 4.

The starting point for both examples is a two-group model. Here we use scripts from these two examples with modifications to adapt the code for characteristics of our data.

We first need to define a suitable model to simulate the true situation in the population. Here we use relevant data from our previous study (Woodhead et al, 2019, 2020) to guide our estimates. This had data from six measures, four of which we plan to use (in slightly modified form) in the current study, two loading on each of the postulated factors. These are Sentence Generation, Phonological Decision, Syntactic Decision and Sentence Comprehension; we term these x1, x2, x4 and x5. The additional two measures, x3 and x6, are selected to act as additional indicators of the language generation and receptive language factors respectively. To simulate data for x1, x2, x4 and x5 we use Session 1 laterality indices (LIs) from the Woodhead et al data, and to simulate x3 we take the mean from Session 2 for x1 and x2; to simulate x6 we take the mean from x4 and x5 from Session 2. This generates a dataset, SimAll, with correlation matrix as shown in Table 1.

##   
## Correlation method: 'pearson'  
## Missing treated using: 'pairwise.complete.obs'

Table 1. Correlations: simulated population model

| rowname | x1 | x2 | x3 | x4 | x5 | x6 |
| --- | --- | --- | --- | --- | --- | --- |
| x1 |  | 0.66 | 0.78 | 0.74 | 0.53 | 0.66 |
| x2 | 0.66 |  | 0.83 | 0.54 | 0.25 | 0.50 |
| x3 | 0.78 | 0.83 |  | 0.74 | 0.45 | 0.68 |
| x4 | 0.74 | 0.54 | 0.74 |  | 0.72 | 0.82 |
| x5 | 0.53 | 0.25 | 0.45 | 0.72 |  | 0.73 |
| x6 | 0.66 | 0.50 | 0.68 | 0.82 | 0.73 |  |

We next need to specify loadings for a population model compatible with our simulated data. Here we first derive loadings from fitting the simulated (SimAll) data.

model.2f <- '  
f1 =~ x1 + x2 + x3   
f2 =~ x4 + x5 +x6   
'  
fit.mod2f <- cfa(model.2f,data=SimAll)  
summary(fit.mod2f, fit.measures=TRUE)

## lavaan 0.6-7 ended normally after 31 iterations  
##   
## Estimator ML  
## Optimization method NLMINB  
## Number of free parameters 13  
##   
## Number of observations 70  
##   
## Model Test User Model:  
##   
## Test statistic 30.748  
## Degrees of freedom 8  
## P-value (Chi-square) 0.000  
##   
## Model Test Baseline Model:  
##   
## Test statistic 367.523  
## Degrees of freedom 15  
## P-value 0.000  
##   
## User Model versus Baseline Model:  
##   
## Comparative Fit Index (CFI) 0.935  
## Tucker-Lewis Index (TLI) 0.879  
##   
## Loglikelihood and Information Criteria:  
##   
## Loglikelihood user model (H0) -676.323  
## Loglikelihood unrestricted model (H1) -660.949  
##   
## Akaike (AIC) 1378.645  
## Bayesian (BIC) 1407.876  
## Sample-size adjusted Bayesian (BIC) 1366.927  
##   
## Root Mean Square Error of Approximation:  
##   
## RMSEA 0.202  
## 90 Percent confidence interval - lower 0.129  
## 90 Percent confidence interval - upper 0.279  
## P-value RMSEA <= 0.05 0.001  
##   
## Standardized Root Mean Square Residual:  
##   
## SRMR 0.082  
##   
## Parameter Estimates:  
##   
## Standard errors Standard  
## Information Expected  
## Information saturated (h1) model Structured  
##   
## Latent Variables:  
## Estimate Std.Err z-value P(>|z|)  
## f1 =~   
## x1 1.000   
## x2 1.180 0.143 8.235 0.000  
## x3 1.170 0.120 9.716 0.000  
## f2 =~   
## x4 1.000   
## x5 0.619 0.074 8.333 0.000  
## x6 0.647 0.058 11.214 0.000  
##   
## Covariances:  
## Estimate Std.Err z-value P(>|z|)  
## f1 ~~   
## f2 2.295 0.521 4.404 0.000  
##   
## Variances:  
## Estimate Std.Err z-value P(>|z|)  
## .x1 1.256 0.234 5.366 0.000  
## .x2 1.234 0.247 4.990 0.000  
## .x3 0.080 0.127 0.630 0.529  
## .x4 0.583 0.231 2.530 0.011  
## .x5 1.151 0.218 5.273 0.000  
## .x6 0.470 0.120 3.924 0.000  
## f1 2.132 0.541 3.941 0.000  
## f2 4.152 0.821 5.056 0.000

standardizedSolution(fit.mod2f)

## lhs op rhs est.std se z pvalue ci.lower ci.upper  
## 1 f1 =~ x1 0.793 0.048 16.534 0.000 0.699 0.887  
## 2 f1 =~ x2 0.840 0.040 21.091 0.000 0.762 0.918  
## 3 f1 =~ x3 0.987 0.022 45.730 0.000 0.944 1.029  
## 4 f2 =~ x4 0.936 0.028 33.762 0.000 0.882 0.991  
## 5 f2 =~ x5 0.762 0.056 13.714 0.000 0.653 0.870  
## 6 f2 =~ x6 0.887 0.034 25.911 0.000 0.820 0.954  
## 7 x1 ~~ x1 0.371 0.076 4.873 0.000 0.222 0.520  
## 8 x2 ~~ x2 0.294 0.067 4.388 0.000 0.163 0.425  
## 9 x3 ~~ x3 0.027 0.043 0.627 0.531 -0.057 0.110  
## 10 x4 ~~ x4 0.123 0.052 2.372 0.018 0.021 0.225  
## 11 x5 ~~ x5 0.420 0.085 4.967 0.000 0.254 0.586  
## 12 x6 ~~ x6 0.213 0.061 3.506 0.000 0.094 0.332  
## 13 f1 ~~ f1 1.000 0.000 NA NA 1.000 1.000  
## 14 f2 ~~ f2 1.000 0.000 NA NA 1.000 1.000  
## 15 f1 ~~ f2 0.772 0.057 13.553 0.000 0.660 0.883

mys<-standardizedSolution(fit.mod2f)

We defined a ‘true’ two-factor population model, H0, using these factor loadings. This will be compared with an alternative, H1, single-factor model. In effect, we ask whether these two models could be differentiated on the basis of the covariance matrix from the simulated data.

# define population model (= H0)  
# Creating the formula using paste is rather confusing, but it does ensure we plug the correct loadings in to the model  
model.pop<-paste0('f1 =~ ',mys[1,4],'\*x1 + ',  
 mys[2,4],'\* x2 + ',  
 mys[3,4],'\* x3   
 f2 =~ ', mys[4,4],'\* x4 + ',  
 mys[5,4],'\* x5 + ',  
 mys[6,4],'\* x6   
 f1~~ 1 \* f1   
 f2~~ 1 \* f2   
 f1~~', mys[15,4],'\*f2')  
  
# define (wrong) H1 model - f1 and f2 perfectly correlated, ie one factor  
model.h1 <- '  
f1 =~ x1 + x2 + x3  
f2 =~ x4 + x5 + x6  
f1~~ 1 \* f1 # variance of f1 is 1  
f2 ~~ 1\*f2  
f1 ~~ 1\*f2  
  
'

Now we make a predicted covariance matrix for the population model and the alternative 1-factor model.

# get population covariance matrix; equivalent to a perfectly fitting H0 model  
cov.h0 <- fitted(sem(model.pop))$cov  
# get covariance matrix as implied by H1 model  
res.h1 <- sem(model.h1, sample.cov = cov.h0, sample.nobs = 1000, likelihood='wishart')  
df <- res.h1@test[[1]]$df  
cov.h1 <- fitted(res.h1)$cov

# perform power analysis specifying 90% power  
ap <- semPower.aPriori(SigmaHat = cov.h1, Sigma = cov.h0, alpha = .05, power = .90, df = df)  
#summary(ap) #displays all content

This analysis estimates that for 90% power to detect superiority of the two-factor over the one-factor model, given true population correlations as in Table 1, we require a minimum sample size of 189.

## Multigroup model (two factor multigroup CFA - left and right handers groups)

This analysis is based on the second extended example in the semPower manual.  
We fit our two-factor model to two groups (left- and right-handers) and are interested in whether the same model estimates can be applied to both groups, or whether fit is better when this invariance requirement is relaxed. There are several potential ways in which the models could differ. On the basis of prior research, we anticipate mean differences in factor scores between left- and right-handers; the question of interest is whether these are sufficient to account for different patterns of laterality. Figure 5 illustrates how a mean shift alone would lead to more cases of opposite laterality in left-handers, even if all factor loadings and covariances in a model were the same.

jpeg('Fig5\_scatters.jpg',width = 700, height = 300)  
par(mfrow=c(1,2))  
require(MASS)  
for (gp in 1:2){ #2 groups, we'll set their means to be 1,1 and 2,2 for simplicity  
mym <- c(gp,gp)  
mysigma <- matrix(c(1,.5,.5,1),nrow=2)  
myN <- 500  
gp1 <- data.frame(mvrnorm(200,mym,mysigma))  
plot(gp1,xlim=c(-2,5),ylim=c(-2,5),pch=16,cex=.7,xlab='LI Factor 1',ylab='LI Factor 2',main=paste0('Group mean = ',gp))  
abline(h=0)  
abline(v=0)  
text(-1,4.8,'A',font=2)  
text(3,4.8,'B',font=2)  
text(-1,-1.8,'C',font=2)  
text(3,-1.8,'D',font=2)  
w1=which(gp1$X1<0)  
w2=which(gp1$X2<0)  
w3=which(gp1$X1>0)  
w4=which(gp1$X2>0)  
quadA <- length(intersect(w1,w4))  
quadB <- length(intersect(w3,w4))  
quadC <- length(intersect(w1,w2))  
quadD <- length(intersect(w2,w3))  
#Percentage in quads A, C and D   
gp2.notLL <- 100\*(quadA+quadC+quadD)/myN  
if(gp==1){gp1.notLL <- 100\*(quadA+quadC+quadD)/myN}  
  
}  
dev.off()

## quartz\_off\_screen   
## 2

Here we compared nested models that restrict parameters iteratively, to test for measurement invariance and then structural invariance between the groups, with mean differences tested at the final step. At each step, if the fit of the latter model is not significantly worse, this supports the conclusion that we can assume that level of measurement invariance.

We first specify a population model for each of the two groups. As before, we will use the SimAll data to derive estimated loadings, this time separately for each handedness group.

cov.sampleR<-cov(SimAll[SimAll$hand=='R',2:7])  
cov.sampleL<-cov(SimAll[SimAll$hand=='L',2:7])  
meansR <- colMeans(SimAll[SimAll$hand=='R',2:7])  
meansL <- colMeans(SimAll[SimAll$hand=='L',2:7])

First we derive indices for a population model for each group by fitting the two factor model to data from our simulated population data, SimAll, this time differentiating by handedness group. We also specify ‘meanstructure’ to allow us to see how means differ between handedness groups.

model.2f <- '  
f1 =~ x1 + x2 + x3   
f2 =~ x4 + x5 + x6   
'  
  
fit.mod2f <- cfa(model.2f,data=SimAll,group="hand", meanstructure=T)  
summary(fit.mod2f, fit.measures=TRUE)

## lavaan 0.6-7 ended normally after 57 iterations  
##   
## Estimator ML  
## Optimization method NLMINB  
## Number of free parameters 38  
##   
## Number of observations per group:   
## R 40  
## L 30  
##   
## Model Test User Model:  
##   
## Test statistic 45.457  
## Degrees of freedom 16  
## P-value (Chi-square) 0.000  
## Test statistic for each group:  
## R 11.051  
## L 34.405  
##   
## Model Test Baseline Model:  
##   
## Test statistic 353.887  
## Degrees of freedom 30  
## P-value 0.000  
##   
## User Model versus Baseline Model:  
##   
## Comparative Fit Index (CFI) 0.909  
## Tucker-Lewis Index (TLI) 0.829  
##   
## Loglikelihood and Information Criteria:  
##   
## Loglikelihood user model (H0) -656.233  
## Loglikelihood unrestricted model (H1) -633.505  
##   
## Akaike (AIC) 1388.467  
## Bayesian (BIC) 1473.909  
## Sample-size adjusted Bayesian (BIC) 1354.214  
##   
## Root Mean Square Error of Approximation:  
##   
## RMSEA 0.229  
## 90 Percent confidence interval - lower 0.152  
## 90 Percent confidence interval - upper 0.309  
## P-value RMSEA <= 0.05 0.000  
##   
## Standardized Root Mean Square Residual:  
##   
## SRMR 0.080  
##   
## Parameter Estimates:  
##   
## Standard errors Standard  
## Information Expected  
## Information saturated (h1) model Structured  
##   
##   
## Group 1 [R]:  
##   
## Latent Variables:  
## Estimate Std.Err z-value P(>|z|)  
## f1 =~   
## x1 1.000   
## x2 1.276 0.278 4.588 0.000  
## x3 1.259 0.252 5.002 0.000  
## f2 =~   
## x4 1.000   
## x5 0.791 0.124 6.397 0.000  
## x6 0.627 0.099 6.307 0.000  
##   
## Covariances:  
## Estimate Std.Err z-value P(>|z|)  
## f1 ~~   
## f2 1.314 0.429 3.060 0.002  
##   
## Intercepts:  
## Estimate Std.Err z-value P(>|z|)  
## .x1 1.685 0.239 7.040 0.000  
## .x2 2.699 0.254 10.633 0.000  
## .x3 2.186 0.214 10.201 0.000  
## .x4 1.068 0.276 3.869 0.000  
## .x5 0.017 0.245 0.068 0.946  
## .x6 0.457 0.196 2.335 0.020  
## f1 0.000   
## f2 0.000   
##   
## Variances:  
## Estimate Std.Err z-value P(>|z|)  
## .x1 1.253 0.303 4.135 0.000  
## .x2 0.887 0.251 3.540 0.000  
## .x3 0.191 0.147 1.299 0.194  
## .x4 0.556 0.243 2.285 0.022  
## .x5 0.840 0.237 3.540 0.000  
## .x6 0.555 0.155 3.590 0.000  
## f1 1.038 0.444 2.337 0.019  
## f2 2.491 0.702 3.549 0.000  
##   
##   
## Group 2 [L]:  
##   
## Latent Variables:  
## Estimate Std.Err z-value P(>|z|)  
## f1 =~   
## x1 1.000   
## x2 1.219 0.198 6.159 0.000  
## x3 1.116 0.149 7.475 0.000  
## f2 =~   
## x4 1.000   
## x5 0.708 0.122 5.785 0.000  
## x6 0.819 0.099 8.266 0.000  
##   
## Covariances:  
## Estimate Std.Err z-value P(>|z|)  
## f1 ~~   
## f2 2.030 0.777 2.611 0.009  
##   
## Intercepts:  
## Estimate Std.Err z-value P(>|z|)  
## .x1 0.302 0.354 0.854 0.393  
## .x2 1.509 0.430 3.506 0.000  
## .x3 0.843 0.342 2.462 0.014  
## .x4 -0.867 0.402 -2.157 0.031  
## .x5 -0.621 0.314 -1.974 0.048  
## .x6 -0.518 0.292 -1.774 0.076  
## f1 0.000   
## f2 0.000   
##   
## Variances:  
## Estimate Std.Err z-value P(>|z|)  
## .x1 1.043 0.324 3.217 0.001  
## .x2 1.523 0.476 3.201 0.001  
## .x3 0.134 0.220 0.609 0.543  
## .x4 1.109 0.377 2.942 0.003  
## .x5 1.090 0.309 3.532 0.000  
## .x6 0.053 0.163 0.326 0.745  
## f1 2.715 0.949 2.860 0.004  
## f2 3.739 1.243 3.008 0.003

A6.z <- standardizedSolution(fit.mod2f) #try using parameters from standardized solution in population model  
  
#We will wrangle this into a table to make it easier to compare groups.  
A6.z$op <- paste(A6.z$lhs,A6.z$op,A6.z$rhs)  
A6.z$lhs[1:23]<-c('path','path','path','path','path','path',  
 'variance','variance','variance','variance','variance','variance','variance','variance',  
 'covariance','mean','mean','mean','mean','mean','mean','mean','mean')  
  
colnames(A6.z)[5:8]<-c('RH: estimate','RH: SE','LH: estimate','LH: SE')  
A6.z[1:23,7:8]<-A6.z[24:46,5:6]  
A6.z<-A6.z[1:23,c(1,2,5:8)]  
A6.z[,3:6]<-round(A6.z[,3:6],3)  
colnames(A6.z)[1:2]<-c('Type','Formula')  
ft<-flextable(A6.z)  
ft <-set\_caption(x=ft,"Two-factor model: estimated loadings for right- and left-handers")  
ft

Two-factor model: estimated loadings for right- and left-handers

| Type | Formula | RH: estimate | RH: SE | LH: estimate | LH: SE |
| --- | --- | --- | --- | --- | --- |
| path | f1 =~ x1 | 0.673 | 0.094 | 0.850 | 0.058 |
| path | f1 =~ x2 | 0.810 | 0.066 | 0.852 | 0.057 |
| path | f1 =~ x3 | 0.947 | 0.044 | 0.981 | 0.032 |
| path | f2 =~ x4 | 0.904 | 0.048 | 0.878 | 0.051 |
| path | f2 =~ x5 | 0.806 | 0.067 | 0.795 | 0.072 |
| path | f2 =~ x6 | 0.799 | 0.068 | 0.990 | 0.032 |
| variance | x1 ~~ x1 | 0.547 | 0.127 | 0.277 | 0.099 |
| variance | x2 ~~ x2 | 0.344 | 0.106 | 0.274 | 0.098 |
| variance | x3 ~~ x3 | 0.104 | 0.083 | 0.038 | 0.063 |
| variance | x4 ~~ x4 | 0.183 | 0.086 | 0.229 | 0.089 |
| variance | x5 ~~ x5 | 0.350 | 0.108 | 0.368 | 0.115 |
| variance | x6 ~~ x6 | 0.362 | 0.109 | 0.021 | 0.064 |
| variance | f1 ~~ f1 | 1.000 | 0.000 | 1.000 | 0.000 |
| variance | f2 ~~ f2 | 1.000 | 0.000 | 1.000 | 0.000 |
| covariance | f1 ~~ f2 | 0.817 | 0.074 | 0.637 | 0.115 |
| mean | x1 ~1 | 1.113 | 0.201 | 0.156 | 0.184 |
| mean | x2 ~1 | 1.681 | 0.246 | 0.640 | 0.200 |
| mean | x3 ~1 | 1.613 | 0.240 | 0.449 | 0.192 |
| mean | x4 ~1 | 0.612 | 0.172 | -0.394 | 0.190 |
| mean | x5 ~1 | 0.011 | 0.158 | -0.360 | 0.188 |
| mean | x6 ~1 | 0.369 | 0.163 | -0.324 | 0.187 |
| mean | f1 ~1 | 0.000 | 0.000 | 0.000 | 0.000 |
| mean | f2 ~1 | 0.000 | 0.000 | 0.000 | 0.000 |

We use the estimated factor loadings and covariance to specify separate population models for the two groups.

#population model group 1   
  
model.pop.R<-paste0('f1 =~ ',A6.z[1,3],'\*x1 + ',  
 A6.z[2,3],'\* x2 + ',  
 A6.z[3,3],'\* x3   
 f2 =~ ', A6.z[4,3],'\* x4 + ',  
 A6.z[5,3],'\* x5 + ',  
 A6.z[6,3],'\* x6   
 x1 ~~ ',(1-A6.z[1,3]^2),'\*x1  
 x2 ~~ ',(1-A6.z[2,3]^2),'\*x2  
 x3 ~~ ',(1-A6.z[3,3]^2),'\*x3  
 x4 ~~ ',(1-A6.z[4,3]^2),'\*x4  
 x5 ~~ ',(1-A6.z[5,3]^2),'\*x5  
 x6 ~~ ',(1-A6.z[6,3]^2),'\*x6  
 f1~~ 1 \* f1   
 f2~~ 1 \* f2   
 f1~~', A6.z[15,3],'\*f2')  
#population model group 2   
model.pop.L<-paste0('f1 =~ ',A6.z[1,5],'\*x1 + ',  
 A6.z[2,5],'\* x2 + ',  
 A6.z[3,5],'\* x3   
 f2 =~ ', A6.z[4,5],'\* x4 + ',  
 A6.z[5,5],'\* x5 + ',  
 A6.z[6,5],'\* x6   
 x1 ~~ ',(1-A6.z[1,5]^2),'\*x1  
 x2 ~~ ',(1-A6.z[2,5]^2),'\*x2  
 x3 ~~ ',(1-A6.z[3,5]^2),'\*x3  
 x4 ~~ ',(1-A6.z[4,5]^2),'\*x4  
 x5 ~~ ',(1-A6.z[5,5]^2),'\*x5  
 x6 ~~ ',(1-A6.z[6,5]^2),'\*x6  
 f1~~ 1 \* f1   
 f2~~ 1 \* f2   
 f1~~', A6.z[15,5],'\*f2')

Now we work out population covariances, assuming separate models for each group.

cov.pop.R <- fitted(sem(model.pop.R))$cov  
cov.pop.L <- fitted(sem(model.pop.L))$cov

## Measurement Invariance

Configural invariance (sometimes known as pattern invariance) ensures that both groups follow the same pattern of factor loadings, i.e. same items load on to the same factors in both groups. Configural invariance is assumed in this study as this was previously established by Woodhead et al. (2020). We start by testing whether factor loadings are the same for each group (weak measurement invariance), which is a necessary prerequisite for interpreting later steps.

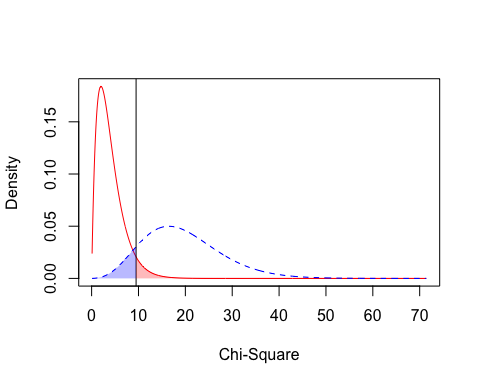
### Weak Measurement Invariance - same loadings (slopes) values for both groups.

Next we define and estimate the first analysis model, setting loadings to be equal - i.e. the alternative model (saturated model with no group constraints) tests whether the fit is as good when the simulated data are tested against a simpler model that has same loadings for both handedness groups.

We can now perform power-analysis for weak measurement invariance.

# perform a priori power analysis  
# Note that lavaan handles multiple group models by having covariance matrices specified as lists containing the associated matrices for each group.   
ap6\_weak <- semPower.aPriori(SigmaHat = list(cov.hcommon.R, cov.hcommon.L),   
 Sigma = list(cov.pop.R, cov.pop.L),   
 alpha = .05, power=.9, N=list(1,1),df = df.diff)  
  
summary(ap6\_weak)

##   
## semPower: A-priori power analysis  
##   
## F0 0.035118   
## RMSEA 0.132511   
## SRMR 0.051992   
## Mc 0.982594   
## GFI 0.988429   
## AGFI 0.939254   
## CFI 0.992222   
##   
## df 4   
## Required Num Observations 442   
## (221, 221)  
##   
## Critical Chi-Square 9.487729   
## NCP 15.45206   
## Alpha 0.050000   
## Beta 0.099015   
## Power (1-beta) 0.900985   
## Implied Alpha/Beta Ratio 0.504973



#NB the N list gives ratio of L hander to R hander

The power analysis gives a minimum sample size of 442. Our intended sample size would not be sufficient to detect the different factor loadings from the simulated data. This is not of concern, as these differences, based on our prior data, are small enough to be disregarded.

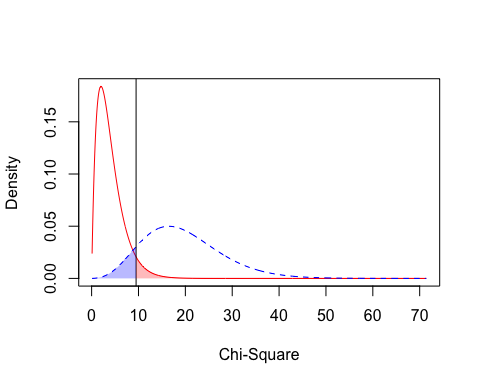
### Strong Measurement Invariance - same loadings (slopes) and item intercept and means values for both groups.

Next we define and estimate the second analysis model, setting loadings and item intercepts to be equal - i.e. the alternative model (loading and intercept constraints) tests whether the fit is as good when the simulated data are tested against a simpler model (loadings only) that has same loadings for both handedness groups.

We can now perform power-analysis for strong measurement invariance.

# perform a priori power analysis  
# Note that lavaan handles multiple group models by having covariance matrices specified as lists containing the associated matrices for each group.   
ap6\_strong <- semPower.aPriori(SigmaHat = list(cov.hcommon.R, cov.hcommon.L),   
 Sigma = list(cov.pop.R, cov.pop.L),   
 alpha = .05, power=.9, N=list(1,1),df = df.diff)  
  
summary(ap6\_strong)

##   
## semPower: A-priori power analysis  
##   
## F0 0.035118   
## RMSEA 0.132511   
## SRMR 0.051992   
## Mc 0.982594   
## GFI 0.988429   
## AGFI 0.939254   
## CFI 0.992222   
##   
## df 4   
## Required Num Observations 442   
## (221, 221)  
##   
## Critical Chi-Square 9.487729   
## NCP 15.45206   
## Alpha 0.050000   
## Beta 0.099015   
## Power (1-beta) 0.900985   
## Implied Alpha/Beta Ratio 0.504973



The power analysis for loadings and intercepts gives a minimum sample size of 442. As before, the analysis shows that any measurement invariance detected in the simulated data is small and would not be detected in our planned sample size.

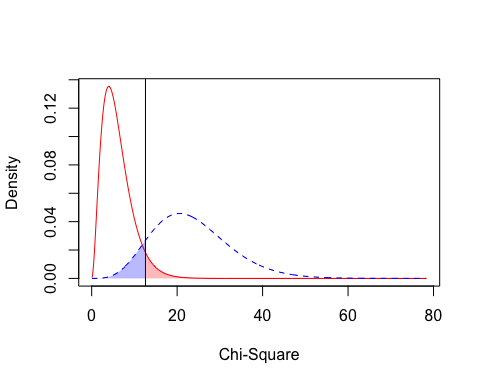
### Strict Measurement Invariance - same loadings, item intercepts and item means, and measurement error values for both groups.

Next we define and estimate the fourth analysis model, setting loadings, item intercepts and means, and measurement errors to be equal - i.e. the alternative model (loadings, item intercepts and means, and errors) tests whether the fit is as good when the simulated data are tested against a simpler model (unconstrained errors) that has same loadings for both handedness groups. This is usually only necessary to ensure that invariance of item reliabilities in both groups.

We can now perform power-analysis for strict measurement invariance.

# perform a priori power analysis  
# Note that lavaan handles multiple group models by having covariance matrices specified as lists containing the associated matrices for each group.   
ap6\_strict <- semPower.aPriori(SigmaHat = list(cov.hcommon.R, cov.hcommon.L),   
 Sigma = list(cov.pop.R, cov.pop.L),   
 alpha = .05, power=.9, N=list(1,1),df = df.diff)  
  
summary(ap6\_strict)

##   
## semPower: A-priori power analysis  
##   
## F0 0.236360  
## RMSEA 0.280690  
## SRMR 0.040365  
## Mc 0.888536  
## GFI 0.926967  
## AGFI 0.744385  
## CFI 0.947654  
##   
## df 6   
## Required Num Observations 76   
## (38, 38)  
##   
## Critical Chi-Square 12.59158  
## NCP 17.49066  
## Alpha 0.050000  
## Beta 0.098621  
## Power (1-beta) 0.901379  
## Implied Alpha/Beta Ratio 0.506993



The power analysis gives a minimum sample size of 76. This suggests we may detect groups differences in residuals between handedness groups with our planned sample size, reflecting the fact that left-handers have higher variance than right-handers on some of the measures.

## Structural Invariance

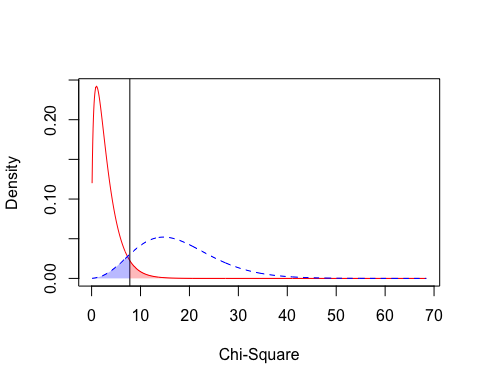
### Factor Variance Invariance Model

Structural invariance is specific to the factors, testing whether firstly factor variances and covariances are invariant across groups, then factor means. We are interested in whether the factor covariances are different across groups so we only test this prediction here

We can now perform power-analysis for structural invariance.

# perform a priori power analysis  
# Note that lavaan handles multiple group models by having covariance matrices specified as lists containing the associated matrices for each group.   
ap6\_structural <- semPower.aPriori(SigmaHat = list(cov.hcommon.R, cov.hcommon.L),   
 Sigma = list(cov.pop.R, cov.pop.L),   
 alpha = .05, power=.9, N=list(1,1),df = df.diff)  
  
summary(ap6\_structural)

##   
## semPower: A-priori power analysis  
##   
## F0 0.262294  
## RMSEA 0.418166  
## SRMR 0.047908  
## Mc 0.877089  
## GFI 0.919598  
## AGFI 0.437188  
## CFI 0.941911  
##   
## df 3   
## Required Num Observations 58   
## (29, 29)  
##   
## Critical Chi-Square 7.814728  
## NCP 14.68847  
## Alpha 0.050000  
## Beta 0.089055  
## Power (1-beta) 0.910945  
## Implied Alpha/Beta Ratio 0.561449



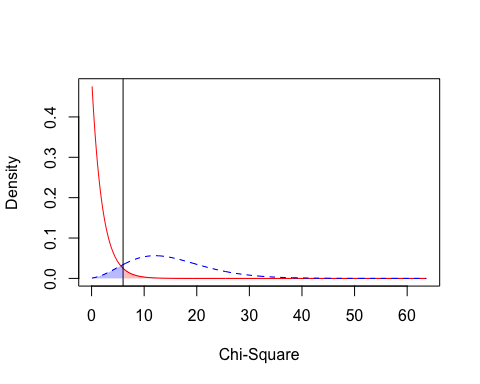
### Factor Mean Invariance Model

Factor mean invariance is specific to the factor means, assuming firstly that factor variances and covariances are invariant across groups. We are interested in whether the factor means are different across groups.

We can now perform power-analysis for structural invariance.

# perform a priori power analysis  
# Note that lavaan handles multiple group models by having covariance matrices specified as lists containing the associated matrices for each group.   
ap6\_structural\_means <- semPower.aPriori(SigmaHat = list(cov.hcommon.R, cov.hcommon.L),   
 Sigma = list(cov.pop.R, cov.pop.L),   
 alpha = .05, power=.9, N=list(1,1),df = df.diff)  
  
summary(ap6\_structural\_means)

##   
## semPower: A-priori power analysis  
##   
## F0 0.262294  
## RMSEA 0.512147  
## SRMR 0.047908  
## Mc 0.877089  
## GFI 0.919598  
## AGFI 0.155782  
## CFI 0.941911  
##   
## df 2   
## Required Num Observations 52   
## (26, 26)  
##   
## Critical Chi-Square 5.991465  
## NCP 13.11470  
## Alpha 0.050000  
## Beta 0.089467  
## Power (1-beta) 0.910533  
## Implied Alpha/Beta Ratio 0.558868



The power analysis gives a minimum sample size of 52.

powertab<-data.frame(Test=c('Weak Invariance','Strong Invariance','Strict Invariance','Structural factor variance Invariance','Structural factor mean Invariance'),N=c(ap6\_weak$requiredN,ap6\_strong$requiredN,ap6\_strict$requiredN,ap6\_structural$requiredN,ap6\_structural\_means$requiredN))  
  
ft<-flextable(powertab)  
ft <- colformat\_num(x = ft,digits = 0)  
ft <- autofit(ft)  
ft <-set\_caption(x=ft,"Sample size required for 90% power")  
ft

Sample size required for 90% power

| Test | N |
| --- | --- |
| Weak Invariance | 442 |
| Strong Invariance | 442 |
| Strict Invariance | 76 |
| Structural factor variance Invariance | 58 |
| Structural factor mean Invariance | 52 |

### Prediction 5

Prediction 5 concerns categorical analysis. A simple approach is to dichotomise laterality at a cutoff of zero for each task, and then perform a chi square analysis to test for association with handedness. For 6 measures, we adopt a Bonferroni-corrected alpha level of .05/6 = .008.

We can first examine sensitivity of this approach with the original sample of Woodhead et al (2020).

This analysis shows that for the language production tasks, the proportions of left- and right-handers who are left-brain lateralised on the language production tasks are close to the proportions that have been obtained with other methods, such as Wada testing: around 90% for right-handers and around 60-70% for left-handers. For the receptive tasks, rates of left-brain bias decrease in both handedness groups, but a handedness effect persists.

Given these results, we aim to achieve a sample size sufficient to detect a difference in left lateralisation of 70% in left-handers vs 90% in right-handers. This difference in percentages translates to an effect size, h, of 0.5157784. Using the R function pwr.2p.test we find that for power = .9 and alpha = .05, minimum N = 79. Thus with the sample size of 100 per handedness group, we would be well-powered to detect this difference.

## Sample size for online test battery

To allow for attrition of participants, we plan to recruit twice as many participants for online testing as we plan to test in person, namely 200 left-handers and 200 right-handers. For testing prediction 1, this will give us 90% power to detect handedness differences in LIs of 0.29 on one-tailed t-test.

For prediction 2 we focus on predicting patterns of covariance between the three online tasks, as we do not have sufficient indicators for modeling with latent variables. We prespecify four possible covariance structures, which are then compared using AIC weights. This is a subset of SEM that does not include any latent variables or directional paths. It allows us to constrain particular covariance patterns and report the ‘best’ model according AIC weights.

The four models include:  
Model A, where all LIs are intercorrelated to a similar degree Model B1, where LIs for the two receptive language measures (dichotic listening and optimal viewing task) are intercorrelated, but independent of rhyme detection Model B2, where LIs for the two tasks involving visual presentation and written language (optimal viewing and rhyme detection) are intercorrelated, and independent of the auditory task, dichotic listening. Model C, where all LIs are independent of one another.

Models B1 and B2 are mathematically equivalent, differing only in the specific variables that are correlated. On a priori grounds, we favour model B1, which is compatible with our overall 2-factor model of language lateralisation. In order to compare models, we need to simulate data corresponding to each model type, and then evaluate how frequently the correct model is identified with our planned sample size of 400 participants.

#========================================================================#  
# simulate multivariate normal data to test ESEM model on three outcomes.#  
#========================================================================#  
#Although we have 4 models, 2 (C and D) are computationally equivalent.  
#In terms of power they will be the same, even though we are making predictions about different sets of tests.  
sim\_data<-function(n=400,r=.5)  
 #we assume tasks are ovp,dichotic, and rhyme  
{  
   
 #========================================================================#  
 sigA <- matrix(c(1, r,r,  
 r, 1, r,  
 r, r, 1),3,3,byrow=TRUE)  
 data\_A <- mvrnorm(n=n, mu = c(0,0,0), Sigma=sigA) #all correlated  
 #========================================================================#  
 sigB <- matrix(c(1, r, 0,  
 r, 1, 0,  
 0, 0, 1),3,3,byrow=TRUE)  
 data\_B <- mvrnorm(n=n, mu = c(0,0,0), Sigma=sigB)  
 #ovp and dichotic correlated, rhyme independent  
 #========================================================================#  
 sigC <- matrix(c(1, 0, 0,  
 0, 1, 0,  
 0, 0, 1),3,3,byrow=TRUE)  
 data\_C <- mvrnorm(n=n, mu = c(0,0,0), Sigma=sigC) #all independent  
 #========================================================================#  
   
   
 data\_A<-as.data.frame(data\_A)  
 data\_B<-as.data.frame(data\_B)  
 data\_C<-as.data.frame(data\_C)  
   
 names(data\_A)<-names(data\_B)<-names(data\_C)<-c("rhyme","dichotic","ovp")  
   
 return(list(data\_A=data\_A,data\_B=data\_B,data\_C=data\_C))  
}  
#========================================================================#

#Specify models  
  
 #Model A all correlated  
 modelA<-"  
 rhyme~~dichotic  
 rhyme~~ovp  
 dichotic~~ovp  
 "  
 #Model B 2 correlated  
 modelB<-"  
 rhyme~~dichotic  
 rhyme~~0\*ovp  
 dichotic~~0\*ovp"  
   
 #model C all independent  
 modelC<-"  
 rhyme~~0\*dichotic  
 rhyme~~0\*ovp  
 dichotic~~0\*ovp"  
  
 modelBB<-"  
 rhyme~~0\*dichotic  
 rhyme~~ovp  
 dichotic~~0\*ovp"

myN <- 400 #sample size  
myr <- .5 #correlation for intercorrelated variables  
mydat<-sim\_data(myN,myr) #simulates sets of data for all 3 models  
domakedata <-0  
niter<-1000  
if (domakedata==1){ #this is slow - can skip if you have already save AICdf  
  
  
AICdf <- data.frame(matrix(NA,nrow=niter,ncol=14))  
colnames(AICdf)<-c('modelA.dataA','modelB.dataA','modelC.dataA',  
 'modelA.dataB','modelB.dataB','modelC.dataB',  
 'modelA.dataC','modelB.dataC','modelC.dataC',  
 'bestfit.dataA','bestfit.dataB','bestfit.dataC',  
 'dataB.sigBvsC','modelBB.dataB')  
   
 for(i in 1:niter)  
 {  
 sims <- sim\_data(n=myN)  
 #Using data simulated from model A (all correlated), test all models  
 fit1A <- cfa(modelA, data=sims$data\_A) #model has zero DF!  
 fit2A <- cfa(modelB, data=sims$data\_A)  
 fit3A <- cfa(modelC, data=sims$data\_A)  
   
 aic\_vector1<- c(fitMeasures(fit1A, "aic"),fitMeasures(fit2A, "aic"),fitMeasures(fit3A, "aic"))  
   
 #Using data simulated from 2-correlated model, test all models  
 #We also test model BB, where 2 are correlated, but the wrong 2  
 fit1B <- cfa(modelA, data=sims$data\_B)#model has zero DF!  
 fit2B <- cfa(modelB, data=sims$data\_B)  
 fit3B <- cfa(modelC, data=sims$data\_B)  
 fit4BB <- cfa(modelBB,data=sims$data\_B)   
   
 aic\_vector2 <- c(fitMeasures(fit1B, "aic"),fitMeasures(fit2B, "aic"),fitMeasures(fit3B, "aic"))  
   
 #Using data simulated from model C (all uncorrelated), test all models  
 fit1C <- cfa(modelA, data=sims$data\_C)#model has zero DF!  
 fit2C <- cfa(modelB, data=sims$data\_C)  
 fit3C <- cfa(modelC, data=sims$data\_C)  
   
 aic\_vector3 <- c(fitMeasures(fit1C, "aic"),fitMeasures(fit2C, "aic"),fitMeasures(fit3C, "aic"))  
   
 aic\_vector4 <-c(fitMeasures(fit4BB,'aic'),fitMeasures(fit2B, "aic"))  
   
 #Anticipate that for each model, lowest AIC obtained when the simulated data do match the model  
 #   
 # The AIC weights make massive difference for preferred model  
 AICdf[i,1:3]<-akaike.weights(aic\_vector1)$weights  
 AICdf[i,4:6]<-akaike.weights(aic\_vector2)$weights  
 AICdf[i,7:9]<-akaike.weights(aic\_vector3)$weights  
 AICdf$bestfit.dataA[i] <- which(AICdf[i,1:3]==max(AICdf[i,1:3]))  
 AICdf$bestfit.dataB[i] <- which(AICdf[i,4:6]==max(AICdf[i,4:6]))  
 AICdf$bestfit.dataC[i] <- which(AICdf[i,7:9]==max(AICdf[i,7:9]))  
   
#If true model is model B, can it be distinguished from model C (all independent)  
 AICdf$dataB.sigBvsC[i] <- anova(fit2B,fit3B)$`Pr(>Chisq)`[2]  
   
 BvsBB<-akaike.weights(aic\_vector4)$weights[1] #comparison of true vs wrong assignment to correlations for model B; if this is < .5, then true model is better.  
 AICdf[i,14]<-ifelse(BvsBB<.5,1,0)  
}  
}  
rr<-round(myr\*100,0)  
AICname<-paste0('AICdf\_r',rr,'niter',niter,'.csv')  
if (domakedata==0){AICdf<-read.csv(AICname)}  
myt1<-table(AICdf$bestfit.dataA)  
myt2<-table(AICdf$bestfit.dataB)  
myt3<-table(AICdf$bestfit.dataC)  
myt4<-table(AICdf$modelBB.dataB)  
powerBC <-length(which(AICdf$dataB.sigBvsC<.05))/niter  
  
if(domakedata==1)  
 {write.csv(AICdf,AICname,row.names=F)}

On 1000 runs, data for 400 participants were simulated to correspond to model B1, with a correlation of .5 between rhyme judgement and OVP, and zero correlation of these LIs with dichotic listening. On 85.6% of runs, model B1 had the highest value of Akaike weights, with model A (all tests correlated) on the other 14.4% of runs. On 100 % of runs, the correct model was significantly superior to model C (zero correlation between LIs), as evidenced by a significant chi square value on likelihood ratio test between models. Model B1 always had higher Akaike weights than model B2, when the data were generated from model B1. On the other hand, if data were generated from model C, ie. a true null model, with zero correlation between the three LIs, then model B1 gave highest Akaike weights on 13% of runs, and model A (all correlated) gave highest Akaike weights on 10.2% of runs.

# References