

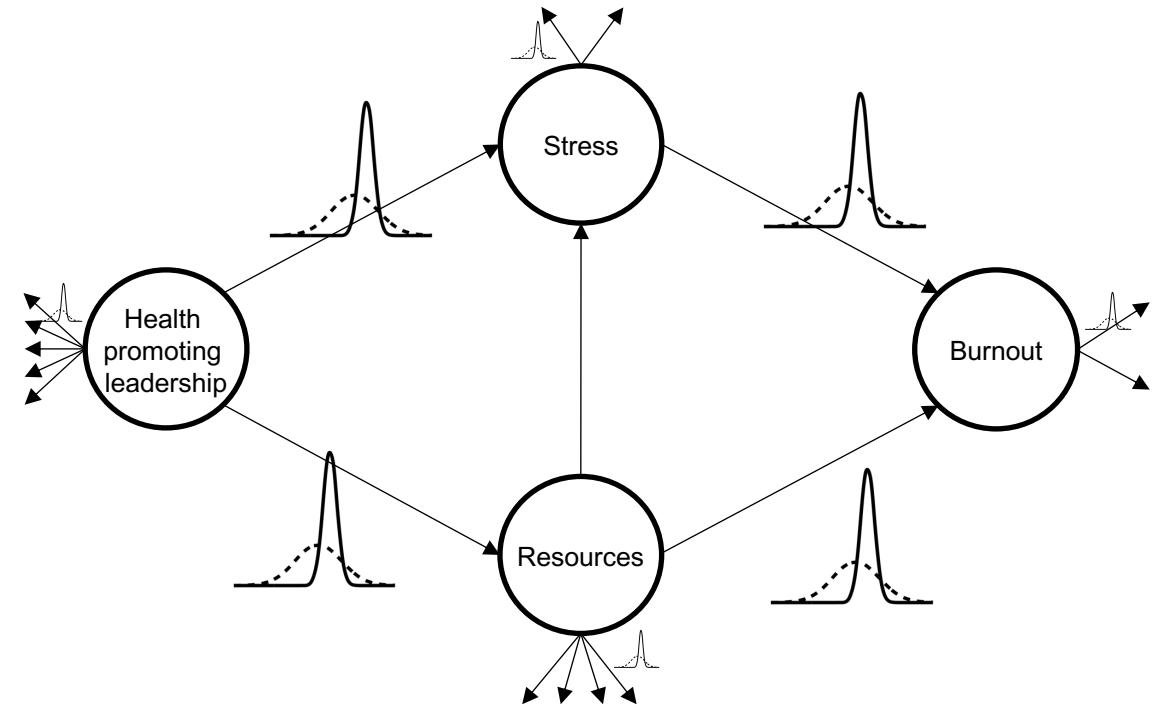
# Structural Equation Modelling

## Lecture 4: Bayesian Structural Equation Modelling

Julius Pfadt

# Recap

- Last week: (Frequentist) SEM
- Today: Bayesian SEM



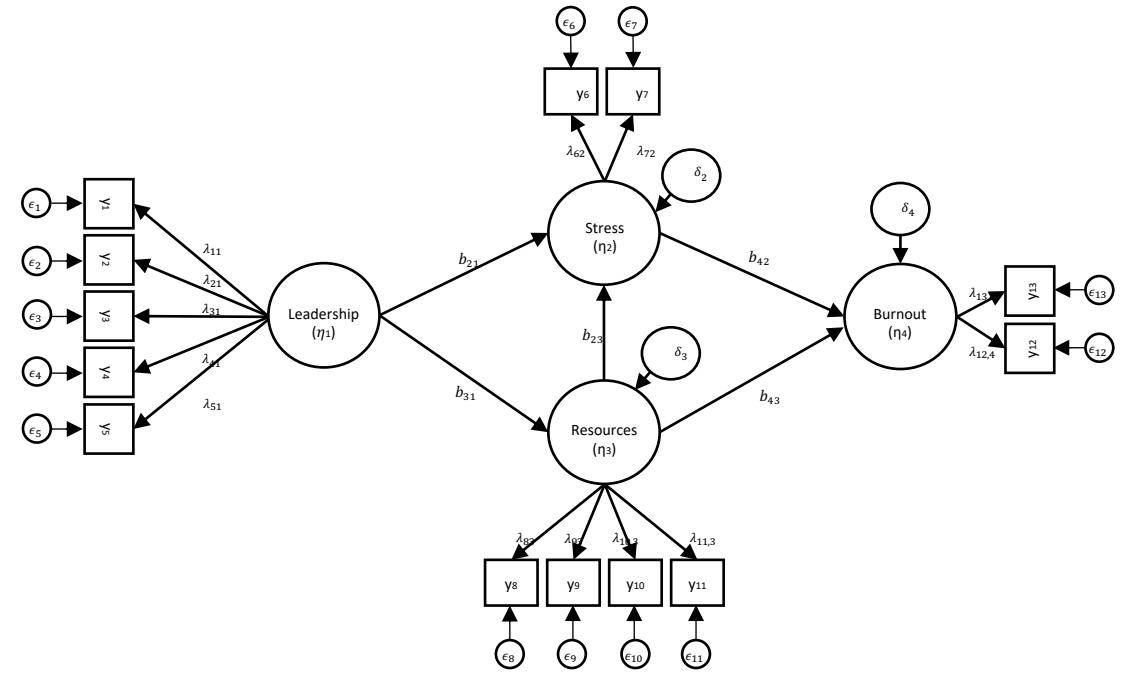
# Introduction

## Interest:

- Parameter estimates: loadings, residual variances, latent regressions, latent variances, latent means
- How well does the hypothesized model represent the data?

## Differences to classical:

- *Fitting* the model
- Model fit is somewhat different
- Identification
- Multiple models should be considered simultaneously

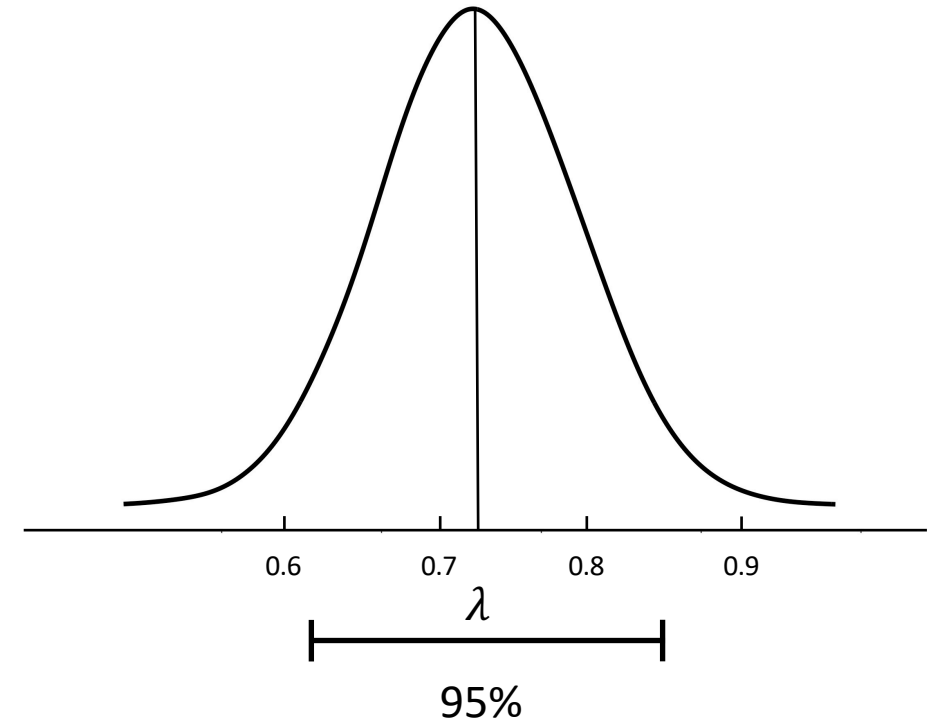


# Outline

- Introduction
- Parameter estimation
  - Basics
  - MCMC sampling
  - Small variance priors
- Model fit and comparison
- Multi-model inference
- Practical issues

# Parameter estimation

- A parameter is a variable with a distribution that we want to approximate, the *posterior*
- The posterior distribution:
  - probability of possible parameter values after observing the data
  - Which parameter values are more likely than others
  - 95% interval contains the parameter of interest with given probability
  - What parameter value is the most likely (point estimate: mean or median)



# Parameter estimation

The posterior distribution of a parameter is proportional to

$$p(\theta|y) \propto p(\theta) \cdot p(y|\theta)$$

With:

$\theta$  as the parameter vector

$y$  as the data

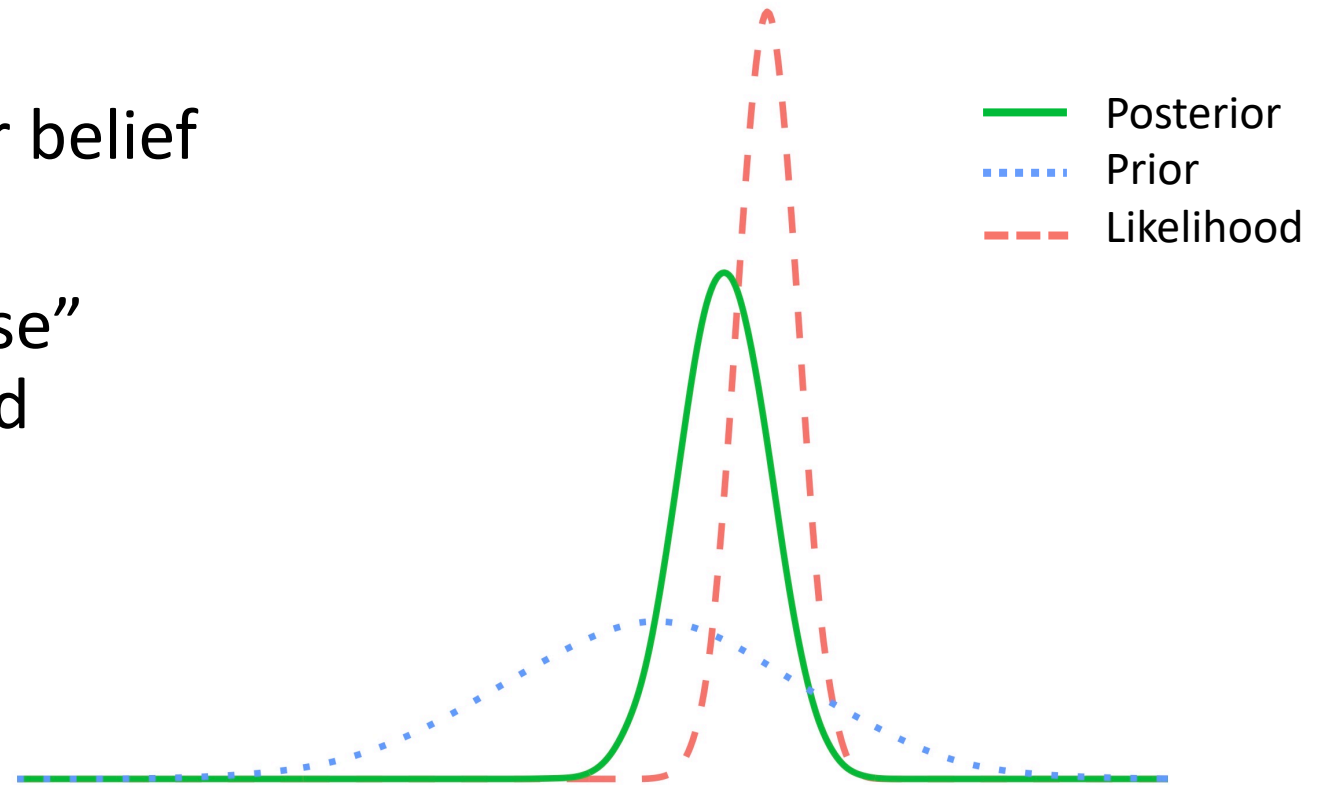
$p(\theta|y)$  as the posterior probability of the parameter given the data

$p(\theta)$  as the prior probability of the parameter

$p(y|\theta)$  as the likelihood of the data given the parameter

# Prior and posterior as probability distributions

- Probability as the mass under the density curve (integral)
- Likelihood changes the prior belief towards posterior belief
- Posterior as the “compromise” between prior and likelihood



# Priors in a SEM

- Prior captures belief about plausible parameter values before seeing the data
- Informative vs. non-informative?

The parameters:

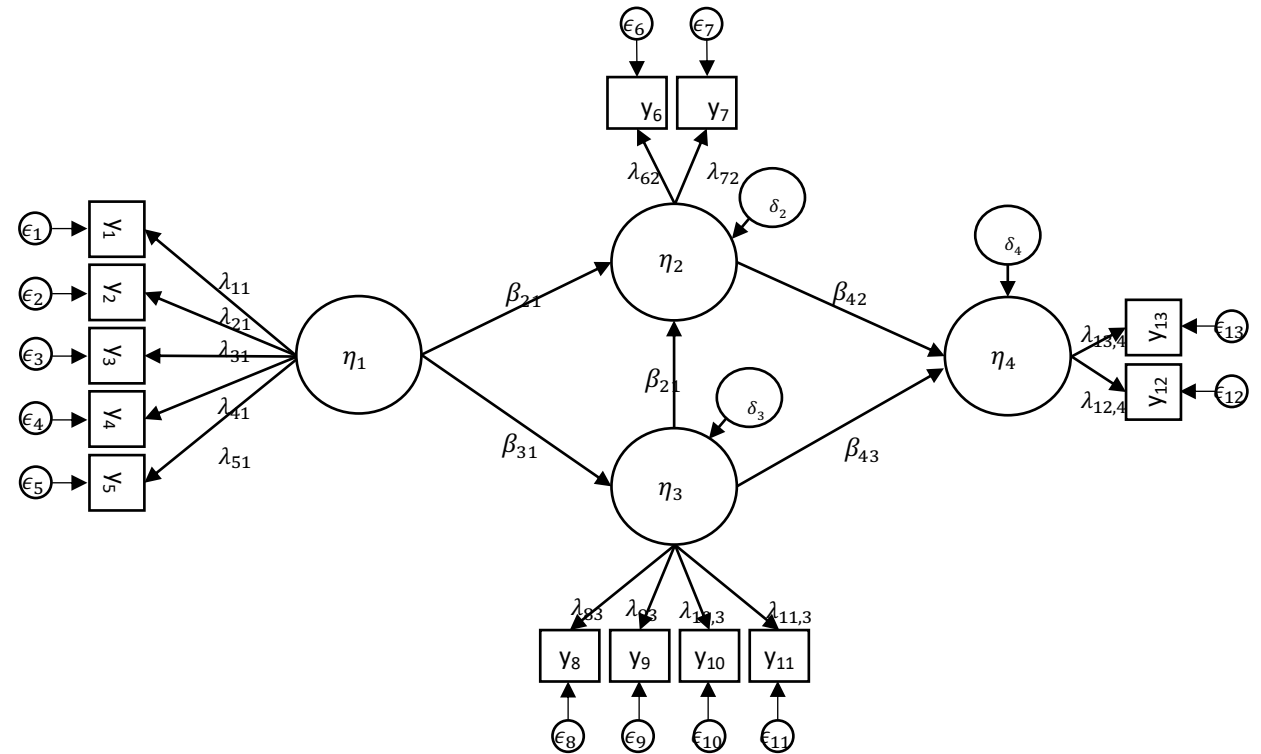
**$\Lambda$** : loadings

**$B$** : regressions

**$\Psi$** : variances and covariances of the latent variables

**$\Theta$** : variances and covariances of the observed residuals

**$\eta$** : latent factor scores





# Posterior estimation

- If we were to write the full joint posterior of all parameters:

$$p(\Lambda, \mathbf{B}, \Psi, \Theta, \eta | \mathbf{y}) \propto p(\Lambda, \mathbf{B}, \Psi, \Theta, \eta) p(\mathbf{y} | \Lambda, \mathbf{B}, \Psi, \Theta, \eta)$$

- An analytic solution to this becomes almost impossible
- Markov-chain Monte Carlo (MCMC) sampling allows to approximate the posterior by sampling from it
- Using software such as JAGS and Stan we could define the priors and the likelihood in the respective programming language and have the built-in MCMC sampling take care of the rest
- But: R-package *lavaan* does this 🤖
- We only need to:
  - Specify a model in *lavaan*-syntax
  - Be aware of what we are doing:
    - MCMC sampling needs to be checked
    - Parameters are no ML point estimates anymore

# MCMC sampling

- Many iterations (samples) to accurately represent the target distribution
- Multiple chains to make sure if the process always leads to the same target and is independent of the starting values
- Burn-in (warm-up) to throw away early samples that are usually not very representative of the target and are more related to the starting value than to the target

```
> library(blavaan)
Loading required package: Rcpp
This is blavaan 0.5-4
On multicore systems, we suggest use of future::plan("multicore") or
future::plan("multisession") for faster post-MCMC computations.
```

```
> library(lavaan)
This is lavaan 0.6-17
lavaan is FREE software! Please report any bugs.
> future::plan("multisession")
> options(mc.cores = parallel::detectCores())
```

```
> model <- "
+ f_leader =~ HealthAwareness + Workload + Reward + Fairness + ValueFit
+ f_resources =~ OverallRecovery + LeisureBreaks + WorkRelatedResources + PsychosocialResources
+ f_stress =~ SocialEmotionalStress + PerformanceRelatedStress
+ f_burnout =~ EmotionalExhaustion + Cynicism
+
+ f_burnout ~ f_stress + f_resources
+ f_stress ~ f_leader + f_resources
+ f_resources ~ f_leader
+
+ "
```

```
> blav_fit <- bsem(model, dtFull, seed = 11)
starting worker pid=26069 on localhost:11188 at 22:50:18.309
starting worker pid=26082 on localhost:11188 at 22:50:18.377
starting worker pid=26095 on localhost:11188 at 22:50:18.447
```

```
SAMPLING FOR MODEL 'stanmarg' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 0.000246 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 2.46 seconds.
Chain 1: Adjust your expectations accordingly!
Chain 1:
Chain 1:
Chain 1: Iteration: 1 / 1500 [ 0%] (Warmup)
```

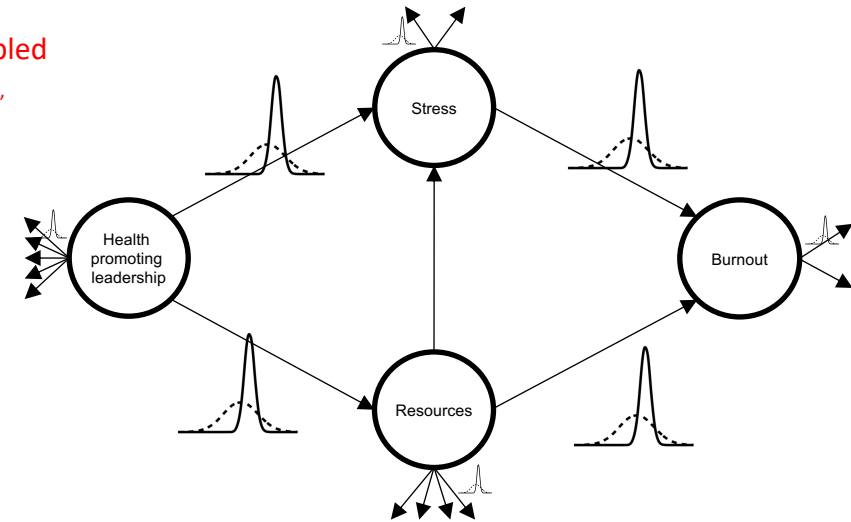
```
SAMPLING FOR MODEL 'stanmarg' NOW (CHAIN 2).
Chain 2:
Chain 2: Gradient evaluation took 0.000226 seconds
Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 2.26 seconds.
Chain 2: Adjust your expectations accordingly!
Chain 2:
```

Call the package and use parallel computing, the chains can be sampled in parallel (these commands are OS dependent, Windows is probably different)

Define the model as usual

Set a seed for reproducibility

Gives us information about the ongoing sampling



> summary(blav\_fit, standardized = TRUE)  
 blavaan 0.5.3.1230 ended normally after 1000 iterations

Estimator BAYES  
 Optimization method MCMC  
 Number of model parameters 31

Number of observations 491

Statistic MargLogLik PPP  
 Value -6788.737 0.000

Marginal likelihood and PPP: Look at this later

Parameter Estimates:

Latent Variables:

	Estimate	Post.SD	pi.lower	pi.upper	Std.lv	Std.all	Rhat	Prior
f_leader =~								
HealthAwareness	1.000				0.873	0.858		
Workload	0.989	0.039	0.916	1.070	0.864	0.862	1.000	normal(0,10)
Reward	1.098	0.036	1.030	1.173	0.959	0.933	1.000	normal(0,10)
Fairness	1.119	0.038	1.048	1.197	0.977	0.938	1.000	normal(0,10)
ValueFit	1.097	0.038	1.024	1.175	0.958	0.919	1.000	normal(0,10)
f_resources =~								
OverallRecovery	1.000				0.814	0.813		
LeisureBreaks	0.815	0.054	0.710	0.923	0.663	0.662	1.000	normal(0,10)
WorkReltdRsrcs	0.969	0.055	0.864	1.080	0.789	0.786	1.000	normal(0,10)
PsychosclRsrcs	0.848	0.055	0.742	0.955	0.691	0.683	1.001	normal(0,10)
f_stress =~								
SocLEmtnlStrss	1.000				0.878	0.873		
PrfrmncRltdStr	1.052	0.043	0.966	1.136	0.924	0.919	1.000	normal(0,10)
f_burnout =~								
EmotionlExhstn	1.000				0.898	0.879		
Cynicism	0.842	0.053	0.740	0.949	0.757	0.733	1.000	normal(0,10)

Rhat: next slide

A normal prior with mean of 0 and SD of 10

Estimate is the mean of the posterior for the parameter, then Posterior SD, posterior 95% interval lower and upper

For comparison: Frequentist results:

Regressions:

	Estimate	Std.Err	z-value	P(> z )	Std.lv	Std.all
f_burnout ~						
f_stress	0.730	0.050	14.516	0.000	0.716	0.716
f_resources	-0.249	0.051	-4.879	0.000	-0.227	-0.227
f_stress ~						
f_leader	-0.009	0.070	-0.124	0.901	-0.009	-0.009
f_resources	-0.594	0.084	-7.104	0.000	-0.554	-0.554
f_resources ~						
f_leader	0.680	0.044	15.434	0.000	0.729	0.729

Regressions:

	Estimate	Post.SD	pi.lower	pi.upper	Std.lv	Std.all	Rhat	Prior
f_burnout ~								
f_stress	0.730	0.054	0.627	0.836	0.714	0.714	0.999	normal(0,10)
f_resources	-0.252	0.053	-0.357	-0.151	-0.228	-0.228	0.999	normal(0,10)
f_stress ~								
f_leader	-0.006	0.072	-0.144	0.139	-0.006	-0.006	1.000	normal(0,10)
f_resources	-0.600	0.084	-0.765	-0.435	-0.556	-0.556	1.001	normal(0,10)
f_resources ~								
f_leader	0.680	0.042	0.597	0.765	0.730	0.730	1.000	normal(0,10)

Variiances:

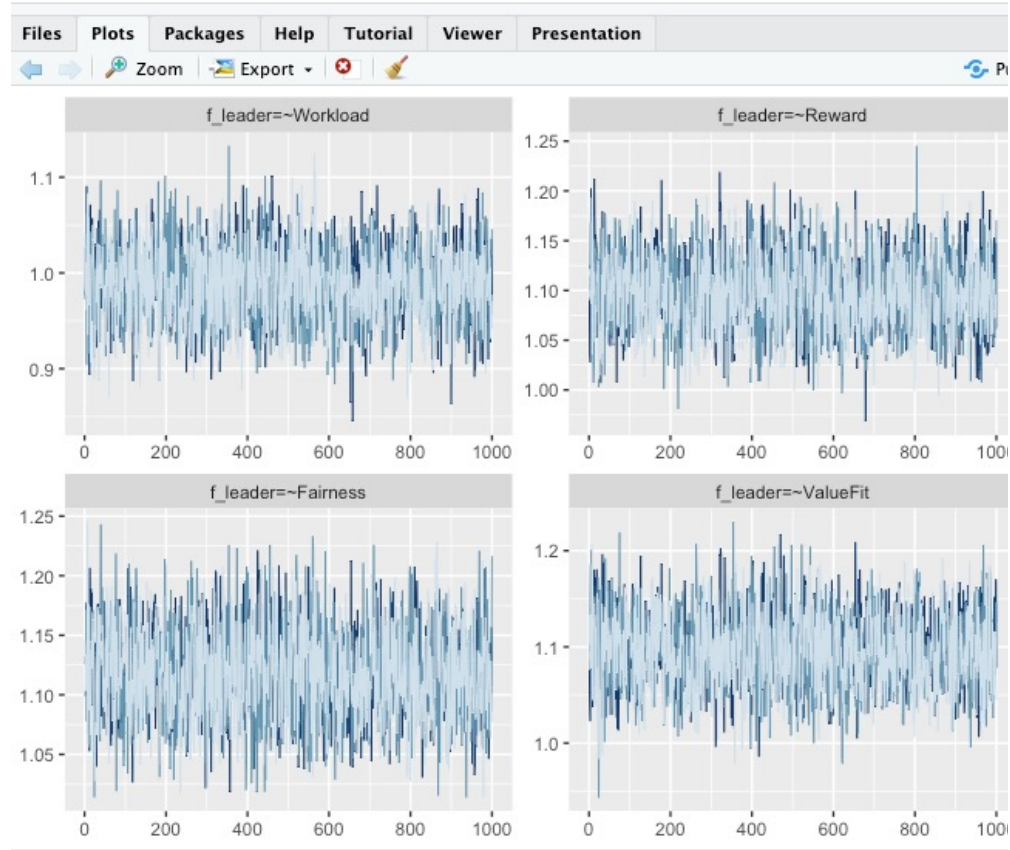
	Estimate	Post.SD	pi.lower	pi.upper	Std.lv	Std.all	Rhat	Prior
.HealthAwareness	0.273	0.020	0.237	0.312	0.273	0.264	1.000	gamma(1, .5)[sd]
.Workload	0.259	0.019	0.224	0.298	0.259	0.258	1.000	gamma(1, .5)[sd]
.Reward	0.136	0.012	0.115	0.161	0.136	0.129	0.999	gamma(1, .5)[sd]

# Convergence diagnostics

Does the MCMC sample properly approximate the target distribution, aka, the posterior?

- R-hat:
  - Similarity of the chains
  - Should be smaller than 1.01 and close to 1.0
- Effective sample size (ESS, neff):
  - Because of autocorrelation MCMC samples not independent
  - ESS is the number of independent sample draws
  - Ideally close to the number of samples
  - *Thinning* can help: Take only every, e.g., 2<sup>nd</sup>, 4<sup>th</sup>, or 10<sup>th</sup> value of a chain
- Traceplots: visualize the chains

```
> plot(blav_fit, pars = 1:4, plot.type = "trace")
> |
```



The traceplots look good: hairy caterpillar

```
> blavInspect(blav_fit, what = "rhat")
```

```
f_leader=~Workload 1.0004983
f_leader=~Reward 1.0008243
f_leader=~Fairness 1.0005618
f_leader=~ValueFit 1.0008286
f_resources=~LeisureBreaks 1.0005187
f_resources=~WorkRelatedResources 0.9998543
f_resources=~PsychosocialResources 0.9992834
f_stress=~PerformanceRelatedStress 1.0010401
f_burnout=~Cynicism 0.9998085
HealthAwareness~~HealthAwareness 1.0001440
Workload~~Workload 1.0003352
Reward~~Reward 1.0005263
Fairness~~Fairness 1.0000491
ValueFit~~ValueFit 0.9993236
OverallRecovery~~OverallRecovery 0.9997663
LeisureBreaks~~LeisureBreaks 0.9995578
WorkRelatedResources~~WorkRelatedResources
```

R-hat values are within the "good" range

```
> blavInspect(blav_fit, what = "neff")
```

```
f_leader=~Workload 2222.686
f_leader=~Reward 1915.312
f_leader=~Fairness 1892.717
f_leader=~ValueFit 2050.289
f_resources=~LeisureBreaks 3409.622
f_resources=~WorkRelatedResources 3215.196
f_resources=~PsychosocialResources 3346.523
f_stress=~PerformanceRelatedStress 2894.546
f_burnout=~Cynicism 3030.250
HealthAwareness~~HealthAwareness 4365.089
Workload~~Workload 4230.501
Reward~~Reward 4738.278
Fairness~~Fairness 4364.824
ValueFit~~ValueFit 4869.630
OverallRecovery~~OverallRecovery 3110.338
```

ESS is good

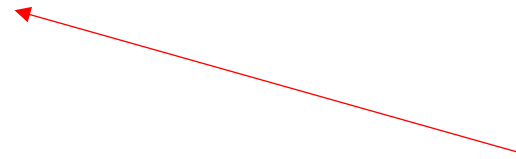
# Priors and sample size

- Explicit benefit of BSEM to work with small samples:
  - Large sample properties of ML are not needed: asymptotical normality
  - Convergence is not needed
  - Prior knowledge can be incorporated
- With large sample sizes, prior does rarely matter: the likelihood will dominate (the prior is “overwhelmed”)
- With smaller samples it can make sense to think a bit longer about the prior, look for prior knowledge
- However: less data, less information

```
> mydp <- dpriors(beta="normal(1,2)")
> blav_fit_pri <- bsem(model, dtFull, seed = 11, dp = mydp)
```

Regressions:

	Estimate	Post.SD	pi.lower	pi.upper	Std.lv	Std.all	Rhat	Prior
f_burnout ~								
f_stress	0.730	0.056	0.621	0.842	0.713	0.713	1.000	normal(1,2)
f_resources	-0.253	0.056	-0.362	-0.146	-0.229	-0.229	1.001	normal(1,2)
f_stress ~								
f_leader	-0.008	0.074	-0.150	0.136	-0.008	-0.008	1.000	normal(1,2)
f_resources	-0.598	0.086	-0.769	-0.432	-0.553	-0.553	1.000	normal(1,2)
f_resources ~								
f_leader	0.681	0.044	0.597	0.769	0.730	0.730	1.000	normal(1,2)

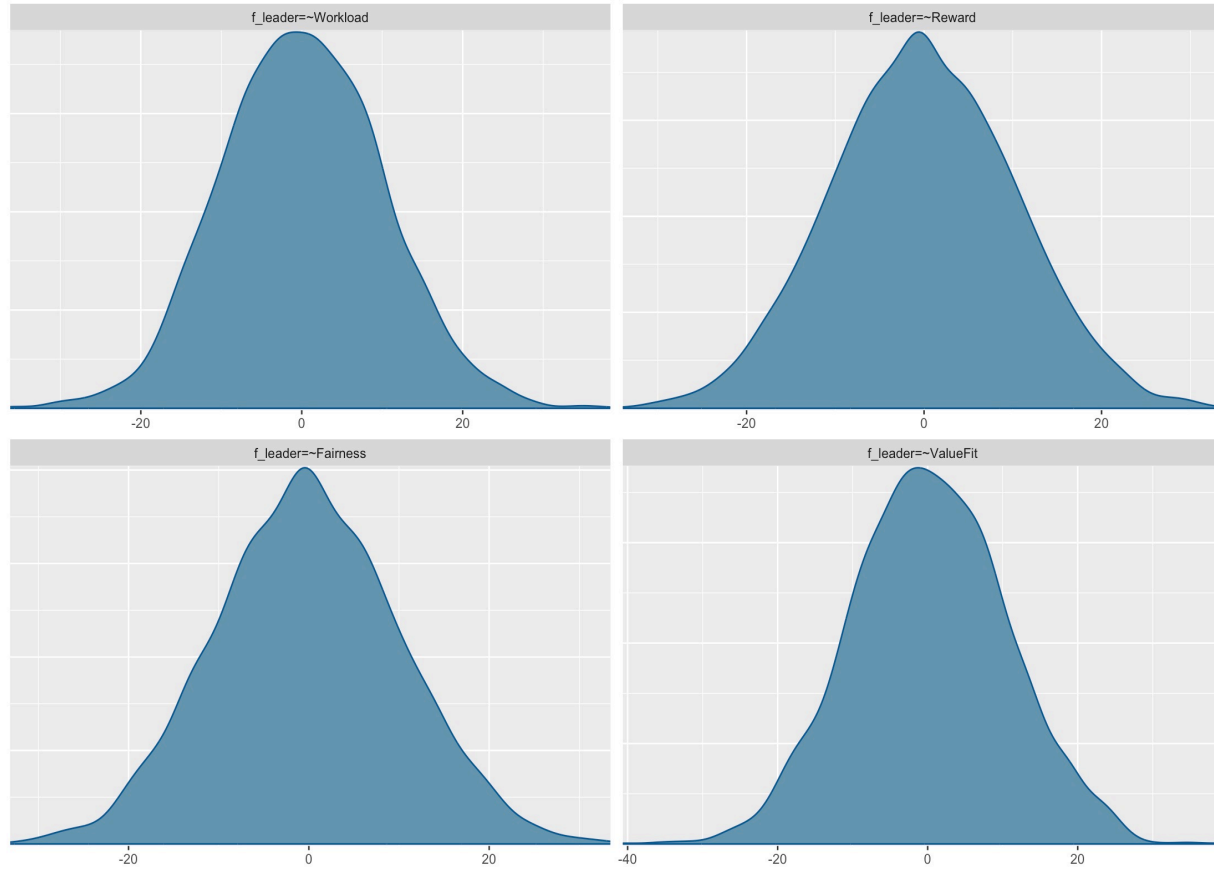


Results barely change



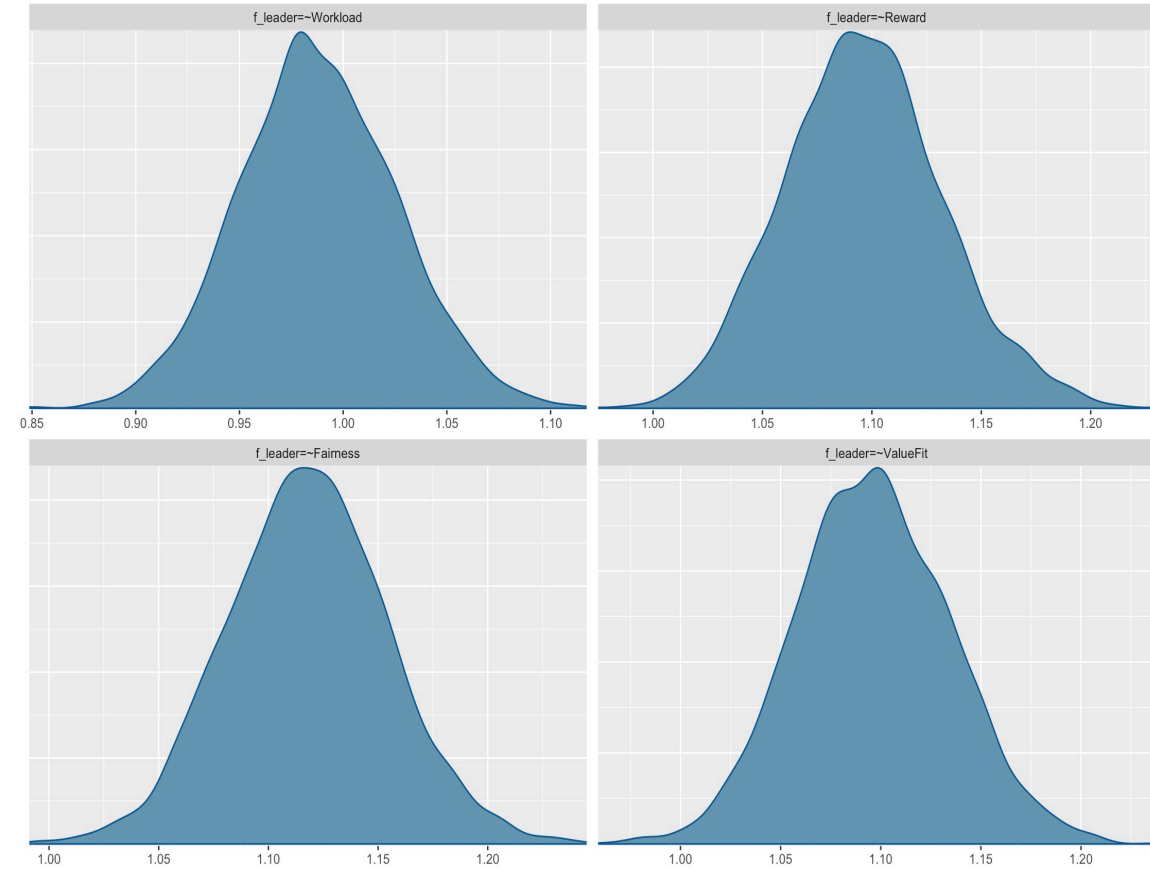
```
blav_fit_only <- bsem(model, dtFull, prisamp = TRUE)
plot(blav_fit_only, pars = 1:4, plot.type = "dens")
```

Samples and plots the  
priors



```
plot(blav_fit, pars = 1:4, plot.type = "dens")
```

Plots the posterior



# Identification and small-variance priors

- Classical identification: Match between data provided information and to be estimated parameters
- In the Bayesian framework a classically unidentified model can be identified
- Instead of relying on the data to provide the information to estimate unidentified parameters, we provide the information ourselves in the form of priors
- This allows us to implement small variance priors or approximate inequality constraints
  - Improves model fit
  - Helps with modification
  - But: May conceal important misspecifications

```
> model_adjusted <- '  
+ f_leader =~ HealthAwareness + Workload + Reward + Fairness + ValueFit  
+ f_resources =~ OverallRecovery + LeisureBreaks + WorkRelatedResources + PsychosocialResources  
+ f_stress =~ SocialEmotionalStress + PerformanceRelatedStress  
+ f_burnout =~ EmotionalExhaustion + Cynicism  
+  
+ f_burnout ~ f_stress + f_resources  
+ f_stress ~ f_leader + f_resources  
+ f_resources ~ f_leader  
+  
+ # small variance prior on a crossloading  
+ f_leader =~ prior("normal(0, 0.08)")*OverallRecovery  
+ '
```

← Add a small variance prior

# Outline

- Introduction
- Parameter estimation
  - Basics
  - MCMC sampling
  - Small variance priors
- Model fit and comparison
- Multi-model inference
- Practical issues

# Model fit

- In general, the idea is:
  - How well does the model implied covariance matrix reproduce the data covariance matrix
- Single model fit:
  - BRMSEA, BCFI, BTLI
  - PPC: posterior predictive (model) checking
    - Posterior predictive p-value (PPP)
- Model comparison:
  - Information criteria: DIC, WAIC, LOOIC
  - Bayes factors

# Model fit – the classics

- The fit function from maximum likelihood estimation

$$F = \log(|\hat{\Sigma}|) + \text{trace}(\mathbf{S}\hat{\Sigma}^{-1}) - \log(|\mathbf{S}|) - k$$

can be seen as a deviance measure for the model implied covariance and the data covariance matrix

- We need the F value in the calculation of RMSEA, CFI, and TLI;
- Posterior sample of implied covariance matrices
  - > posterior sample of F-values
  - > posterior sample of RMSEA, CFI, and TLI: BRMSEA, BCFI, BTLI
- We also need the model complexity
- Model complexity used to be number of parameters  $p$
- Because of the priors the model complexity is not equal to  $p$  in the Bayesian framework, it is an estimated quantity

```
> model_null <- "  
+ HealthAwareness ~~ HealthAwareness  
+ Workload ~~ Workload  
+ Control ~~ Control  
+ Reward ~~ Reward  
+ Community ~~ Community  
+ Fairness ~~ Fairness  
+ ValueFit ~~ ValueFit  
+ OverallRecovery ~~ OverallRecovery  
+ LeisureBreaks ~~ LeisureBreaks  
+ WorkRelatedResources ~~ WorkRelatedResources  
+ PsychosocialResources ~~ PsychosocialResources  
+ SocialEmotionalStress ~~ SocialEmotionalStress  
+ PerformanceRelatedStress ~~ PerformanceRelatedStress  
+ LossOfMeaning ~~ LossOfMeaning  
+ EmotionalExhaustion ~~ EmotionalExhaustion  
+ Cynicism ~~ Cynicism  
+ "  
> blav_fit_null <- bsem(model = model_null, data = dtFull)
```

For calculation of CFI and TLI we need the null (baseline) model

We already needed that in the frequentist SEM, but lavaan did that under the hood

```
> blavFitIndices(blav_fit, baseline.model = blav_fit_null, fit.measures = c("BRMSEA", "BCFI", "BTLI"))  
Posterior mean (EAP) of devm-based fit indices:
```

BRMSEA	BCFI	BTLI
0.116	0.949	0.936

# PPP

```
> summary(blav_fit, standardized = TRUE)
blavaan 0.5.3.1230 ended normally after 1000 iterations

Estimator              BAYES
Optimization method    MCMC
Number of model parameters  44

Number of observations  491

Statistic              MargLogLik
Value                  -6878.963
```

Statistic	MargLogLik	PPP
Value	-6878.963	0.000

- Posterior predictive p-value
- Compare the discrepancy of:
  - Posterior of model implied covariance matrices and observed covariance matrix
  - Simulated data from the posterior covariance matrices and the model implied covariance matrices (chance discrepancy, sampling distribution, a “good” fit discrepancy)
- Does the model’s discrepancy differ significantly from the discrepancy expected by chance?
- Should be 0.5, the smaller the worse



# Information criteria

- Deviance information criterion (DIC)
  - Point estimate of the deviance of the mean model implied covariance matrix
- Widely applicable information criterion (WAIC)
  - The deviance for each person for each posterior sample
- Leave-one-out cross validation information criterion (LOOIC):
  - Deviance between the model trained for N-1 data and the remaining one observation as the test data
  - N-times
  - for MCMC of SEM computationally very expensive
  - Approximation obtained with importance sampling (*blavaan* does that for you)

```

> model_adjusted <- '
+ f_leader =~ HealthAwareness + Workload + Reward + Fairness + ValueFit
+ f_resources =~ OverallRecovery + LeisureBreaks + WorkRelatedResources + PsychosocialResources
+ f_stress =~ SocialEmotionalStress + PerformanceRelatedStress
+ f_burnout =~ EmotionalExhaustion + Cynicism
+
+ f_burnout ~ f_stress + f_resources
+ f_stress ~ f_leader + f_resources
+ f_resources ~ f_leader
+
+ # residual covariance
+ WorkRelatedResources ~ Cynicism
+ '
> blav_fit_adjusted <- bsem(model_adjusted, dtFull)

```

Fit a model with an additional residual covariance



```

> fitmeasures(blav_fit)
  npar    logl    ppp    bic    dic    p_dic    waic    p_waic    se_waic    looic    p_loo    se_loo margloglik
31.000 -6650.763  0.000 13493.552 13363.379  30.927 13365.286  32.509  129.768 13365.364  32.548  129.771 -6788.655
Warning message:

```

5 (1.0%) p\_waic estimates greater than 0.4. We recommend trying loo instead.

Ignore the warning

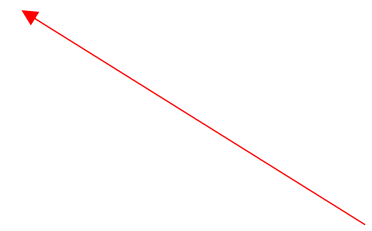


```

> fitmeasures(blav_fit_adjusted)
  npar    logl    ppp    bic    dic    p_dic    waic    p_waic    se_waic    looic    p_loo    se_loo margloglik
32.000 -6615.039  0.000 13428.300 13294.149  32.035 13296.331  33.827  128.700 13296.429  33.876  128.704      NA
Warning message:

```

6 (1.2%) p\_waic estimates greater than 0.4. We recommend trying loo instead.



```

> blavCompare(blav_fit, blav_fit_adjusted)

```

```

WAIC estimates:
object1: 13365.286
object2: 13296.331

```

```

ELPD difference & SE:
-34.478  8.453

```

```

LOO estimates:
object1: 13365.363
object2: 13296.43

```

```

ELPD difference & SE:
-34.466  8.453

```

```

Laplace approximation to the log-Bayes factor
(experimental; positive values favor object1):  NA

```

The various information criteria show lower values for the adjusted model, we consider BF later



# Frequentist model comparison

- Compare **two** models
- $H_0$ : the models are equal
- We can only reject the  $H_0$

But

- How much more likely is the complex model than the simple model?
- What is the probability of each model given the data?
- Within the space of possible models, which is the most likely?

# Bayes factor

- Comparison of two models  $M_1$  and  $M_2$  by comparing their marginal likelihoods (the probability of the data under the model)
- In parameter estimation:  $p(y|M) = p(y)$
- The ratio of the likelihoods is the Bayes factor:

$$BF_{12} = \frac{p(y|M_1)}{p(y|M_2)}$$

- How much more likely are the data under  $M_1$  than under  $M_2$ ?

# Bayes factor

- Can quantify evidence in favour of  $M_1$  but also in favour of  $M_2$
- For example:
  - $BF_{12} = 3$  -> the data are three times as likely under  $M_1$  than under  $M_2$
  - How much more likely are the data under  $M_2$  than under  $M_1$ ?  $BF_{21} = 1/BF_{12}$
- For SEM the marginal likelihoods are not easy to estimate:
  - Approximations exist: Laplace approximation, BIC transformation for the BF
  - You could use sampling methods such as importance sampling, path sampling, bridgesampling

TABLE 15.1: The Bayes factor scale as proposed by Jeffreys (1939). This scale should not be regarded as a hard and fast rule.

$BF_{12}$	Interpretation
$> 100$	Extreme evidence for $\mathcal{M}_1$ .
$30 - 100$	Very strong evidence for $\mathcal{M}_1$ .
$10 - 30$	Strong evidence for $\mathcal{M}_1$ .
$3 - 10$	Moderate evidence for $\mathcal{M}_1$ .
$1 - 3$	Anecdotal evidence for $\mathcal{M}_1$ .
1	No evidence.
$\frac{1}{1} - \frac{1}{3}$	Anecdotal evidence for $\mathcal{M}_2$ .
$\frac{1}{3} - \frac{1}{10}$	Moderate evidence for $\mathcal{M}_2$ .
$\frac{1}{10} - \frac{1}{30}$	Strong evidence for $\mathcal{M}_2$ .
$\frac{1}{30} - \frac{1}{100}$	Very strong evidence for $\mathcal{M}_2$ .
$< \frac{1}{100}$	Extreme evidence for $\mathcal{M}_2$ .

<https://vasishth.github.io/bayescogsci/book/ch-bf.html>

```
> model_adjusted <- '
+ f_leader =~ HealthAwareness + Workload + Reward + Fairness + ValueFit
+ f_resources =~ OverallRecovery + LeisureBreaks + WorkRelatedResources + PsychosocialResources
+ f_stress =~ SocialEmotionalStress + PerformanceRelatedStress
+ f_burnout =~ EmotionalExhaustion + Cynicism
+
+ f_burnout ~ f_stress + f_resources
+ f_stress ~ f_leader + f_resources
+ f_resources ~ f_leader
+
+ # crossloading
+ f_leader =~ OverallRecovery
+ '
> blav_fit_adjusted <- bsem(model_adjusted, dtFull)
```

We specify a model with a crossloading

```
> mll1 <- fitmeasures(blav_fit, "margloglik")
> mll2 <- fitmeasures(blav_fit_adjusted, "margloglik")
>
```

Obtain the marginal log likelihood, the marginal likelihood itself would be a number so small R would just say 0

```
> # logbf12
> mll1 - mll2
```

We continue with the log values, then the ratio becomes a difference: obtain the logBF12, which is very small, so lots of evidence against M1

```
margloglik
-25.784
```

```
> # logbf21
> mll2 - mll1
```

```
margloglik
25.784
```

The inverse, so logBF21 is very large, so lots of evidence for M2

```
> blavCompare(blav_fit, blav_fit_adjusted)
```

```
WAIC estimates:
object1: 13365.195
object2: 13300.147
```

```
ELPD difference & SE:
-32.524 7.800
```

```
L00 estimates:
object1: 13365.266
object2: 13300.245
```

```
ELPD difference & SE:
-32.511 7.800
```

```
Laplace approximation to the log-Bayes factor
(experimental; positive values favor object1): -25.784
```

# Outline

- Introduction
- Parameter estimation
  - Basics
  - MCMC sampling
  - Small variance priors
- Model fit and comparison
- Multi-model inference
- Practical issues

# Bayesian model averaging (BMA)

- With BFs we are still comparing one model against another, eventually settling on one “good” model which we base all inferences on
- What if we estimated the posterior model probability (PMP) of each possible model?
- We could use the PMP as weights for our parameter estimates



# BMA

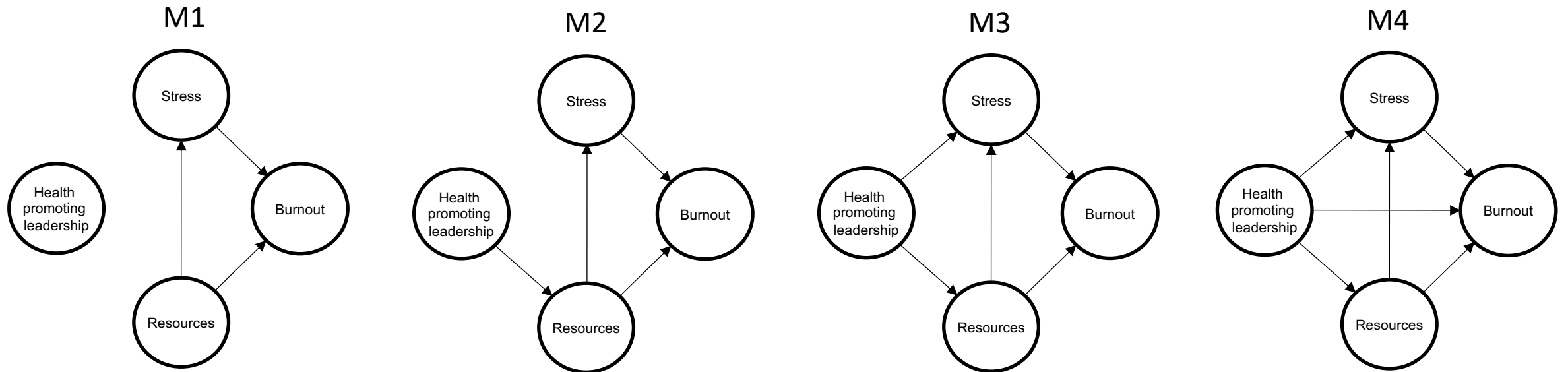
- Lets say we have 4 candidate models
- What is the posterior model probability of M1?

$$p(M_1|y) = \frac{p(M_1) \cdot p(y|M_1)}{\sum_{j=1}^4 p(y|M_j)p(M_j)}$$

- In practice we often choose a uniform prior for the models, that is, all models are equally likely a-priori, so  $p(M_1) = .25$

# Candidate model

- Choosing candidate models can be hard if the number of possible models is large, let's only look at the regressions for now:



And many more... but let's assume we define only these four as candidate models



# BMA

With the PMPs we can:

- Estimate the posterior inclusion probability of the parameter
  - Sum the posterior model probabilities that include the parameter
- Estimate a model-averaged posterior distribution for a parameter:
  - Draw a model based on the PMPs
  - Draw a value for the parameter from its posterior under this model
  - Repeat many times
  - Doing this for only the models that include the parameter answers the question, assuming the effect is present how strong is it?
- Much more: Inclusion BF, Exclusion BF

# Outline

- Introduction
- Parameter estimation
  - Basics
  - MCMC sampling
  - Small variance priors
- Model fit and comparison
- Multi-model inference
- Practical issues

# Practical Issues for BSEM (and BMA)

- The marginal likelihood approximation is not very stable for complex models or small samples
- The priors are important for BFs and BMA
- Reducing the number of possible candidate models is not straightforward
- Comparing certain models in SEM in a BMA framework is debatable: For example, the latent variable(s) in a one-factor and two-factor model have a different meaning -> comparing 🍏 and 🍊?
- *blavaan* allows a lot of models, but not the same functionality as lavaan
- In general, lots of models do not fit well, BSEM cannot help, well calibrated test instruments and good theory are key

# Last slide

Good luck with the final assignment and hopefully you can enjoy the free time after 😊