

Oravecz & Muth (2017)

# Fitting growth curve models in the Bayesian framework

*Psychonomic Bulletin and Review*

<https://doi.org/10.3758/s13423-017-1281-0>

<https://git.psu.edu/zzo1/FittingGCMBayesian>

<https://github.com/teebusch/jc-bayesgcm>

paper

accompanying git repository

git repository with my code (and these slides)

# General Process of Bayesian Analysis

- Parameters are specified as random variables with a probability distribution (the **prior distribution**).
- We derive the **posterior distribution** of the parameters (i.e. their updated probability distribution) by conditioning on our data.
- We summarize the posterior distribution to make inferences.
- Different priors express different levels of certainty (Objective / vague vs. informative)  
→ more or less influence on posterior distribution

# Motivation for Bayesian Data Analysis

- incorporate previous findings and background knowledge as prior information
- additional options for analysis (e.g. of small sample sizes, single case studies)
- Explore posterior distribution rather than point estimates and sampling distributions (CI's and p-values) → more intuitive
- Independent of sampling intentions / data collection plan / stopping rule
- Very flexible and powerful → Easily(?) accommodate multilevel data, binomial data, outliers, ...

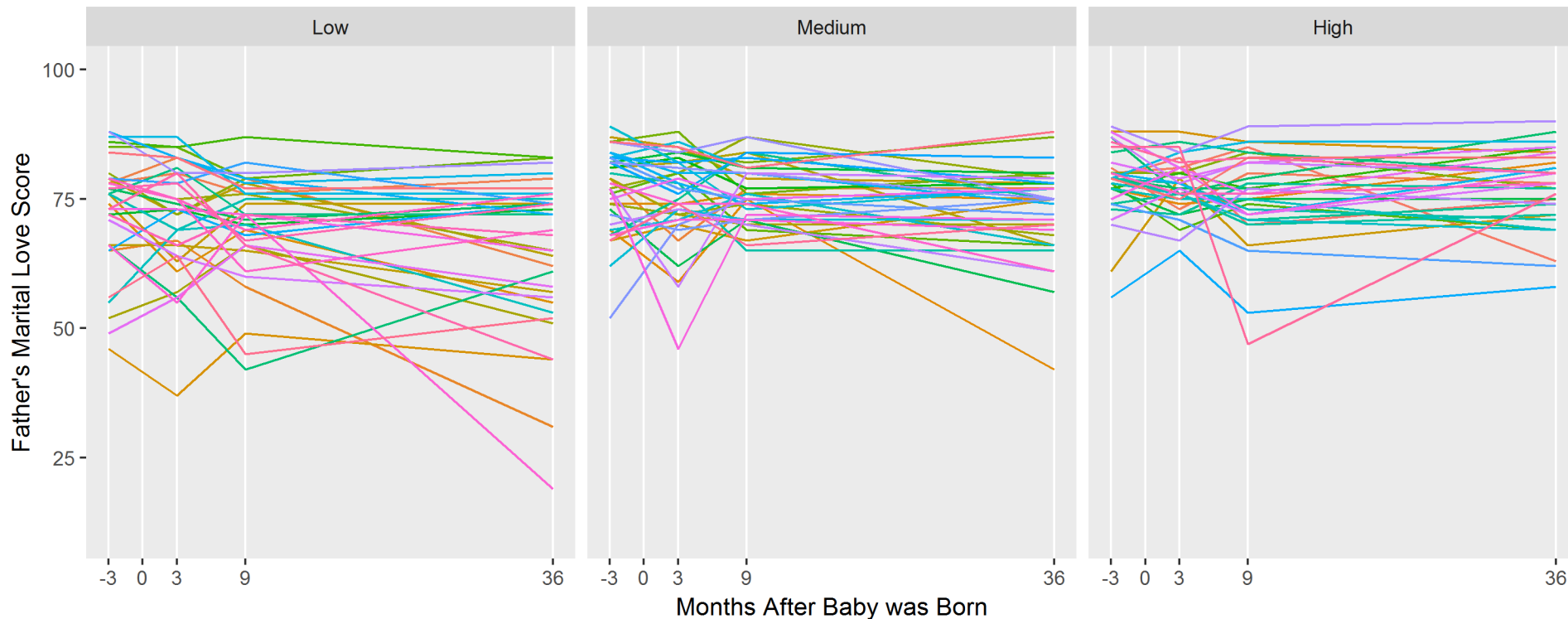
# Growth Curve Models (GCM)

- Special case of Generalized Linear Mixed Model (GLMM)
- Longitudinal Model: quantifies within-person change and between person variability
- Model individual trajectories over time
- Account for dependency (correlation) in repeated measures
- Handles unbalanced designs and missing observations
- Here: Linear GCM, but extension is easy(?)

(Note: GCMs can also be fit in MLE framework, e.g. using lme4 package)

# Example data set

- Changes in father's (n=106) feeling of marital love during transition into parenthood (score between 0-100)
- 4 measurement moments (-3, 3, 9, 36 months after birth of first child)
- groups: life experiences within marriage before child is born (low, medium, high positivity; e.g. job promotion, death of family member... )



# Model Specification

**Level 1:** score of father  $i$  at measurement  $t$

$$Y_{i,t} \sim N(\beta_{i,1} + \beta_{i,2}T_t, \sigma_{e_{Level1}}^2)$$

**Level 2:** Intercept and slope vary by group (low/med./high).

Variance terms capture individual differences between fathers.

Distributions of slopes and intercepts co-vary (bivariate normal distribution).

$$\begin{bmatrix} \beta_{i,1} \\ \beta_{i,2} \end{bmatrix} \sim N_2 \left( \begin{bmatrix} \mu_{MedPInt} + \beta_{lowPInt}X_{i,1} + \beta_{highPInt}X_{i,2} \\ \mu_{MedPSlope} + \beta_{lowPSlope}X_{i,1} + \beta_{highPSlope}X_{i,2} \end{bmatrix}, \begin{bmatrix} \sigma_{e_{\beta_1}}^2 & \sigma_{e_{\beta_{12}}} \\ \sigma_{e_{\beta_{21}}} & \sigma_{e_{\beta_2}}^2 \end{bmatrix} \right)$$

**Note:** person-level estimates are constrained by population trend (i.e. estimation of individual betas incorporates both group-level and individual information)

**Level 1:** score of father i at measurement t

# Model Specification

$$Y_{i,t} \sim N(\beta_{i,1} + \beta_{i,2}T_t, \sigma_{e_{Level1}}^2)$$

**Level 2:** Intercept and slope vary by group (low/med./high).

Variance terms capture individual differences between fathers.

Distributions of slopes and intercepts co-vary (bivariate normal distribution).

$$\begin{bmatrix} \beta_{i,1} \\ \beta_{i,2} \end{bmatrix} \sim N_2 \left( \begin{bmatrix} \mu_{MedPInt} + \beta_{lowPInt}X_{i,1} + \beta_{highPInt}X_{i,2} \\ \mu_{MedPSlope} + \beta_{lowPSlope}X_{i,1} + \beta_{highPSlope}X_{i,2} \end{bmatrix}, \begin{bmatrix} \sigma_{e_{\beta_1}}^2 & \sigma_{e_{\beta_{12}}} \\ \sigma_{e_{\beta_{21}}} & \sigma_{e_{\beta_2}}^2 \end{bmatrix} \right)$$

**Prior Information:** diffuse distributions over wide range of possible values

The prior distribution of the betas is **hyper-parameterized**, i.e. its parameters have their own priors, instead of being fixed.

$$\mu_{MedPInt} \sim N(0, 100)$$

$$\mu_{MedPSlope} \sim N(0, 100)$$

$$\beta_{lowPInt} \sim N(0, 100)$$

$$\beta_{highPInt} \sim N(0, 100)$$

$$\beta_{lowPSlope} \sim N(0, 100)$$

$$\beta_{highPSlope} \sim N(0, 100)$$

$$\sigma_{e_{Level2}} \sim \text{unif}(0, 100)$$

$$\sigma_{e_{\beta_1}} \sim \text{unif}(0, 100)$$

$$\sigma_{e_{\beta_2}} \sim \text{unif}(0, 100)$$

$$\rho_{e_{\beta_{12}}} \sim \text{unif}(-1, 1).$$

$$\sigma_{e_{\beta_{12}}} = \sigma_{e_{\beta_{21}}} = \rho_{e_{\beta_{12}}} \sigma_{e_{\beta_1}} \sigma_{e_{\beta_2}}$$

Error in the paper?  
Shouldn't it be  $\sigma_{e_{Level1}}$ ?

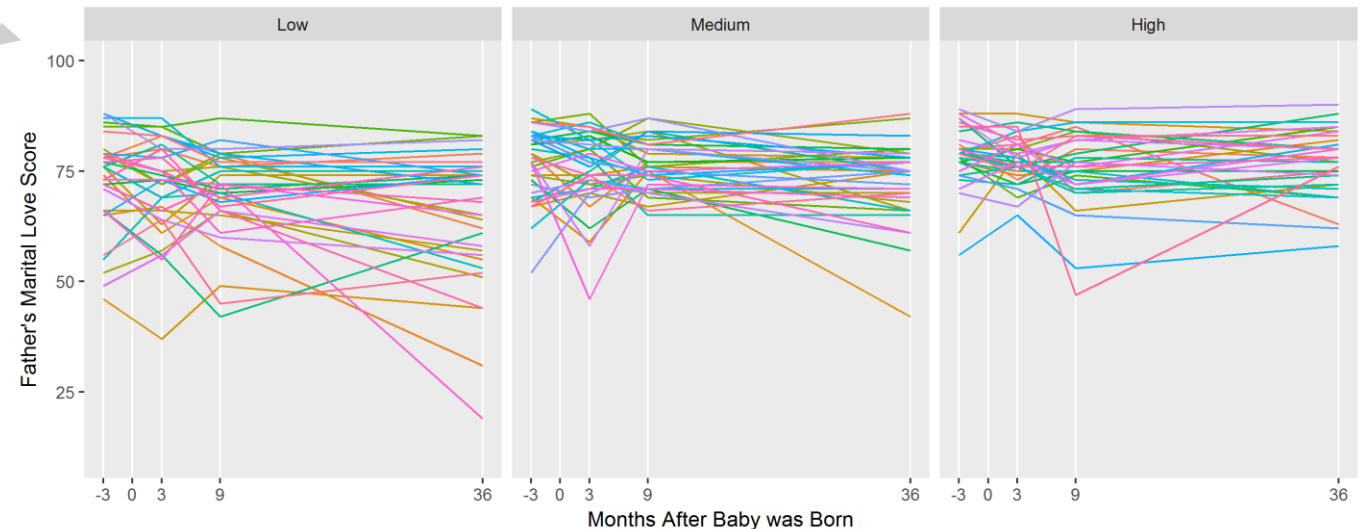
The covariance of slopes and intercepts ( $\sigma_{e_{\beta_{12}}}, \sigma_{e_{\beta_{21}}}$ ) is determined by their r and sd's

# Preparations

- Set up R, JAGS (the Bayesian estimator engine), and rjags
- load the data

	subject	posFactor	lowPos	highPos	month	score
1	1	High	0	1	-3	79
2	1	High	0	1	3	83
3	1	High	0	1	9	70
4	1	High	0	1	36	75
...						
13	4	Low	1	0	-3	78
14	4	Low	1	0	3	80
15	4	Low	1	0	9	76
16	4	Low	1	0	36	79
...						

Note: code in paper is sometimes unnecessarily complex. See <https://github.com/teebusch/jc-bayesgcm> for my super efficient code ;)





# Modeling

R's formula syntax does **not** work with JAGS

e.g. `lme4::lmer(score ~ posFactor*month + (1 + month|subject), data = loveData)`

Instead we have to...

- Pass all data as named list to JAGS

```
jagsData <- list(  
  "nObservations" = nrow(loveData), # number of observations  
  "nSubjects" = length(unique(loveData$subject)), # number of subjects (fathers)  
  "subject" = loveData$subject, # subject id of observation  
  "time" = loveData$month, # month of observation  
  "lowPos" = loveData$lowPos, # dummy var. for low/medium/high positivity factor  
  "highPos" = loveData$highPos,  
  "score" = loveData$score # dv: father's marital love score  
)
```

- and define the model in the “BUGS” modeling language...

```

# Loop over observations
for (i in 1:nObservations) {
  # Likelihood Function (assumed data generating distribution), Eq. 1
  score[i] ~ dnorm(betas[subject[i],1] + betas[subject[i],2]*time[i], 1/pow(sdLevel1Error,2))
}

# Loop over subjects
for (j in 1:nSubjects) {
  # Level 2 bivariate distribution of intercepts and slopes, Eq. 6
  betas[j,1:2] ~ dmnorm(level2MeanVector[j,1:2], interpersonPrecisionMatrix[1:2,1:2])

  level2MeanVector[j,1] <- medPInt + betaLowPInt*lowPos[j] + betaHighPInt*highPos[j]
  level2MeanVector[j,2] <- medPSlope + betaLowPSlope*lowPos[j] + betaHighPSlope*highPos[j]
}

# Prior distributions
medPInt ~ dnorm(0,0.01)
medPSlope ~ dnorm(0,0.01)

betaLowPInt ~ dnorm(0,0.01)
betaLowPSlope ~ dnorm(0,0.01)
betaHighPInt ~ dnorm(0,0.01)
betaHighPSlope ~ dnorm(0,0.01)

sdLevel1Error ~ dunif(0,100)
sdIntercept ~ dunif(0,100)
sdSlope ~ dunif(0,100)
corrIntSlope ~ dunif(-1,1)

# Transforming model parameters
interpersonCovMatrix[1,1] = pow(sdIntercept,2)
interpersonCovMatrix[2,2] = pow(sdSlope,2)
interpersonCovMatrix[1,2] = corrIntSlope * sdIntercept * sdSlope
interpersonCovMatrix[2,1] = interpersonCovMatrix[1,2]
interpersonPrecisionMatrix <- inverse(interpersonCovMatrix)

```

```
# Loop over observations
for (i in 1:nObservations) {
  # Likelihood Function (assumed data generating distribution), Eq. 1
  score[i] ~ dnorm(betas[subject[i],1] + betas[subject[i],2]*time[i], 1/pow(sdLevel1Error,2))
}
```

$$Y_{i,t} \sim N(\beta_{i,1} + \beta_{i,2}T_t, \sigma_{e_{Level1}}^2)$$

```
##### Intercept and slopes, Eq. 6
betas[j,1:2] ~ dnmnorm(level2MeanVector[j,1:2], interpersonPrecisionMatrix[1:2,1:2])

level2MeanVector[j,1] <- medPInt + betaLowPInt*lowPos[j] + betaHighPInt*highPos[j]
level2MeanVector[j,2] <- medPSlope + betaLowPSlope*lowPos[j] + betaHighPSlope*highPos[j]
}

# Prior distributions
medPInt ~ dnorm(0,0.01)
medPSlope ~ dnorm(0,0.01)

betaLowPInt ~ dnorm(0,0.01)
betaLowPSlope ~ dnorm(0,0.01)
betaHighPInt ~ dnorm(0,0.01)
betaHighPSlope ~ dnorm(0,0.01)

sdLevel1Error ~ dunif(0,100)
sdIntercept ~ dunif(0,100)
sdSlope ~ dunif(0,100)
corrIntSlope ~ dunif(-1,1)

# Transforming model parameters
interpersonCovMatrix[1,1] = pow(sdIntercept,2)
interpersonCovMatrix[2,2] = pow(sdSlope,2)
interpersonCovMatrix[1,2] = corrIntSlope * sdIntercept * sdSlope
interpersonCovMatrix[2,1] = interpersonCovMatrix[1,2]
interpersonPrecisionMatrix <- inverse(interpersonCovMatrix)
```

For historical reasons, JAGS uses precision instead of variance to parameterize distributions. Precision is simply the inverse of the variance ( $1/\sigma^2$ )

```

# Loop over observations
for (i in 1:nObservations) {
  # Likelihood Function (assumed data generating distribution), Eq. 1
  score[i] ~ dnorm(betas[subject[i],1] + betas[subject[i],2]*time[i], 1/pow(sdLevel1Error,2))
}

# Loop over subjects
for (j in 1:nSubjects) {
  # Level 2 bivariate distribution of intercepts and slopes, Eq. 6
  betas[j,1:2] ~ dmnorm(level2MeanVector[j,1:2], interpersonPrecisionMatrix[1:2,1:2])

  level2MeanVector[j,1] <- medPInt + betaLowPInt*lowPos[j] + betaHighPInt*highPos[j]
  level2MeanVector[j,2] <- medPSlope + betaLowPSlope*lowPos[j] + betaHighPSlope*highPos[j]
}

```

$$\begin{bmatrix} \beta_{i,1} \\ \beta_{i,2} \end{bmatrix} \sim N_2 \left( \begin{bmatrix} \mu_{\text{MedPInt}} + \beta_{\text{lowPInt}} X_{i,1} + \beta_{\text{highPInt}} X_{i,2} \\ \mu_{\text{MedPSlope}} + \beta_{\text{lowPSlope}} X_{i,1} + \beta_{\text{highPSlope}} X_{i,2} \end{bmatrix}, \begin{bmatrix} \sigma_{e\beta_1}^2 & \sigma_{e\beta_{12}} \\ \sigma_{e\beta_{21}} & \sigma_{e\beta_2}^2 \end{bmatrix} \right)$$

```

betaLowPSlope ~ dnorm(0,0.01)
betaHighPInt ~ dnorm(0,0.01)
betaHighPSlope ~ dnorm(0,0.01)

```

```

sdLevel1Error ~ dunif(0,100)
sdIntercept ~ dunif(0,100)
sdSlope ~ dunif(0,100)
corrIntSlope ~ dunif(-1,1)

```

```

# Transforming model parameters
interpersonCovMatrix[1,1] = pow(sdIntercept,2)
interpersonCovMatrix[2,2] = pow(sdSlope,2)
interpersonCovMatrix[1,2] = corrIntSlope * sdIntercept * sdSlope
interpersonCovMatrix[2,1] = interpersonCovMatrix[1,2]
interpersonPrecisionMatrix <- inverse(interpersonCovMatrix)

```

```
# Loop over observations
for (i in 1:nObservations) {
  # Likelihood Function (assumed data generating distribution), Eq. 1
  score[i] ~ dnorm(betas[subject[i],1] + betas[subject[i],2]*time[i], 1/pow(sdLevel1Error,2))
}
```

```
# Loop over subjects
for (j in 1:nSubjects) {
  # Level 2 bivariate distribution of intercept and slope
  betas[j,1:2] ~ dmnorm(level2MeanVector, level2CovMatrix[1:2,1:2])

  level2MeanVector[j,1] <- medPInt + betaLowPInt*highPos[j]
  level2MeanVector[j,2] <- medPSlope + betaLowPSlope*highPos[j]
}
```

```
# Prior distributions
medPInt ~ dnorm(0,0.01)
medPSlope ~ dnorm(0,0.01)
```

```
betaLowPInt ~ dnorm(0,0.01)
betaLowPSlope ~ dnorm(0,0.01)
betaHighPInt ~ dnorm(0,0.01)
betaHighPSlope ~ dnorm(0,0.01)
```

```
sdLevel1Error ~ dunif(0,100)
sdIntercept ~ dunif(0,100)
sdSlope ~ dunif(0,100)
corrIntSlope ~ dunif(-1,1)
```

```
# Transforming model parameters
interpersonCovMatrix[1,1] = pow(sdIntercept,2)
interpersonCovMatrix[2,2] = pow(sdSlope,2)
interpersonCovMatrix[1,2] = corrIntSlope * sdIntercept * sdSlope
interpersonCovMatrix[2,1] = interpersonCovMatrix[1,2]
interpersonPrecisionMatrix <- inverse(interpersonCovMatrix)
```

$$\mu_{\text{MedPInt}} \sim N(0, 100)$$

$$\mu_{\text{MedPSlope}} \sim N(0, 100)$$

$$\beta_{\text{lowPInt}} \sim N(0, 100)$$

$$\beta_{\text{highPInt}} \sim N(0, 100)$$

$$\beta_{\text{lowPSlope}} \sim N(0, 100)$$

$$\beta_{\text{highPSlope}} \sim N(0, 100)$$

$$\sigma_{e_{\text{Level2}}} \sim \text{unif}(0, 100)$$

$$\sigma_{e_{\beta_1}} \sim \text{unif}(0, 100)$$

$$\sigma_{e_{\beta_2}} \sim \text{unif}(0, 100)$$

$$\rho_{e_{\beta_{12}}} \sim \text{unif}(-1, 1).$$

$$\sigma_{e_{\beta_{12}}} = \sigma_{e_{\beta_{21}}} = \rho_{e_{\beta_{12}}} \sigma_{e_{\beta_1}} \sigma_{e_{\beta_2}}$$

We also add some additional lines to keep track of other parameters of interest (not obligatory for the model to run, but useful for us)

```
# High and low positivity slopes and intercepts
lowPInt <- medPInt + betaLowPInt
highPInt <- medPInt + betaHighPInt
lowPSlope <- medPSlope + betaLowPSlope
highPSlope <- medPSlope + betaHighPSlope

# planned comparisons - contrasts between low, mid, and high intercepts and slopes
c_highLowPInt <- betaHighPInt - betaLowPInt
c_medLowPInt <- - betaLowPInt
c_highMedPInt <- betaHighPInt
c_highLowPSlope <- betaHighPSlope - betaLowPSlope
c_medLowPSlope <- - betaLowPSlope
c_highMedPSlope <- betaHighPSlope
```

Finally, this code has to be saved in a file which is then read by `jags.model()`

Next we estimate our parameters through sampling from their posterior distribution via the MCMC algorithm implemented in JAGS

The diagram illustrates the components of the JAGS model initialization and sampling process. It consists of two main sections of R code, with vertical lines connecting specific arguments to their descriptions.

**Initialization Code:**

```
# initialize model
mdl <- jags.model("loveModel.bugs", jagsData, n.chains = 6, n.adapt = 2000)

# run some burn-in iterations
update(mdl, n.iter = 1000)
```

**Sampling Code:**

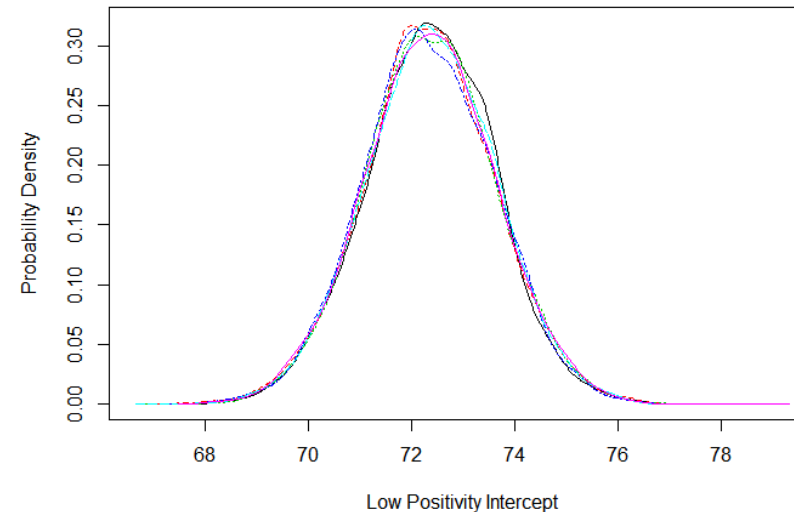
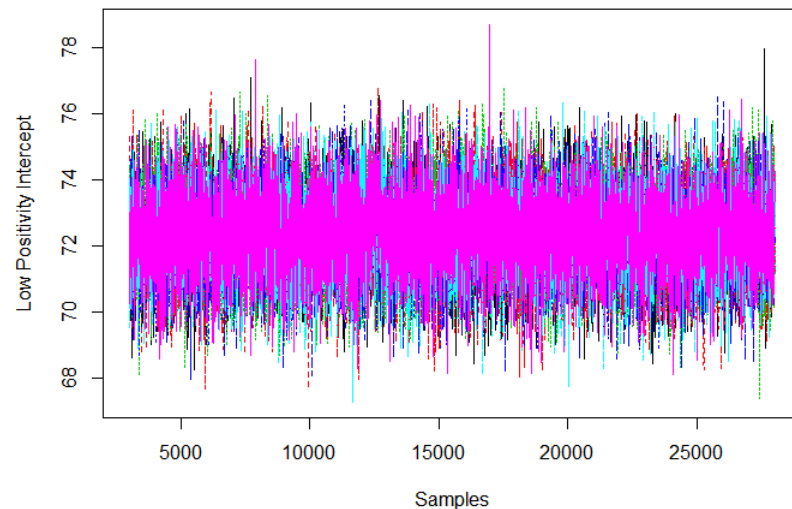
```
# start sampling from the posterior distribution
codaSamples <- coda.samples(mdl, variable.names = parameters, n.iter = 25000, thin = 5)
```

**Descriptions:**

- Text file with model specification (points to "loveModel.bugs")
- Named list with data as defined above (points to jagsData)
- Number of Markov chains to run (points to n.chains = 6)
- Iterations that algorithm uses to adapt to model and data (points to n.adapt = 2000)
- Vector with names of parameters to extract on each iteration (not shown) (points to variable.names = parameters)
- Total number of iterations (points to n.iter = 25000)
- Thinning interval to decrease autocorrelation (use only every 5<sup>th</sup> sample) (points to thin = 5)

# Model Checks

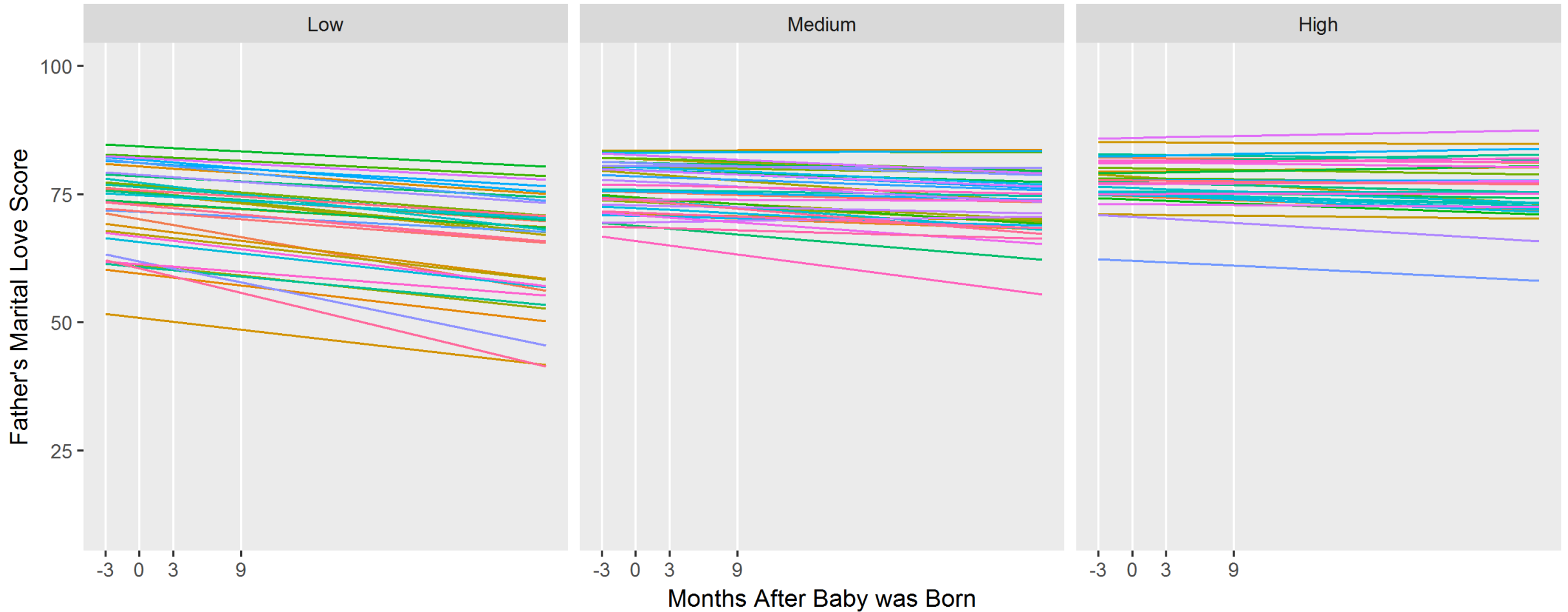
- For each parameter we ran 6 chains with different starting values. Have all converged to the same area, i.e. do all chains find similar likely values for the parameter?
- **Graphical check (repeat for all parameters):**



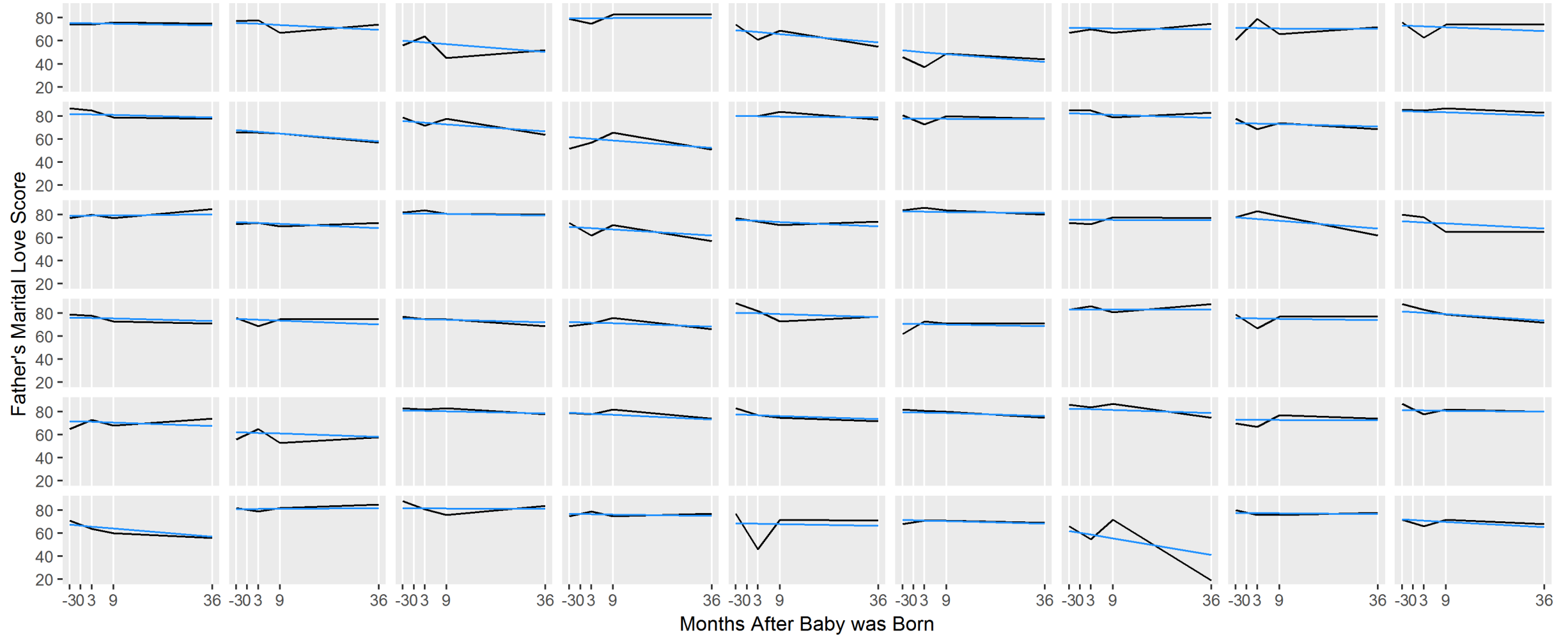
- **Numerical checks** (not shown here):  
 $\hat{R} < 1.1$  for all parameters? → Gelman-Rubin convergence diagnostic: ratio of between and within chain variances. Check e.g. with `gelman.diag(codaSamples, multivariate = FALSE)`  
 $ESS \approx 10,000$  for all parameters → Effective sample size ("n good" samples, i.e. no autocorrelation)



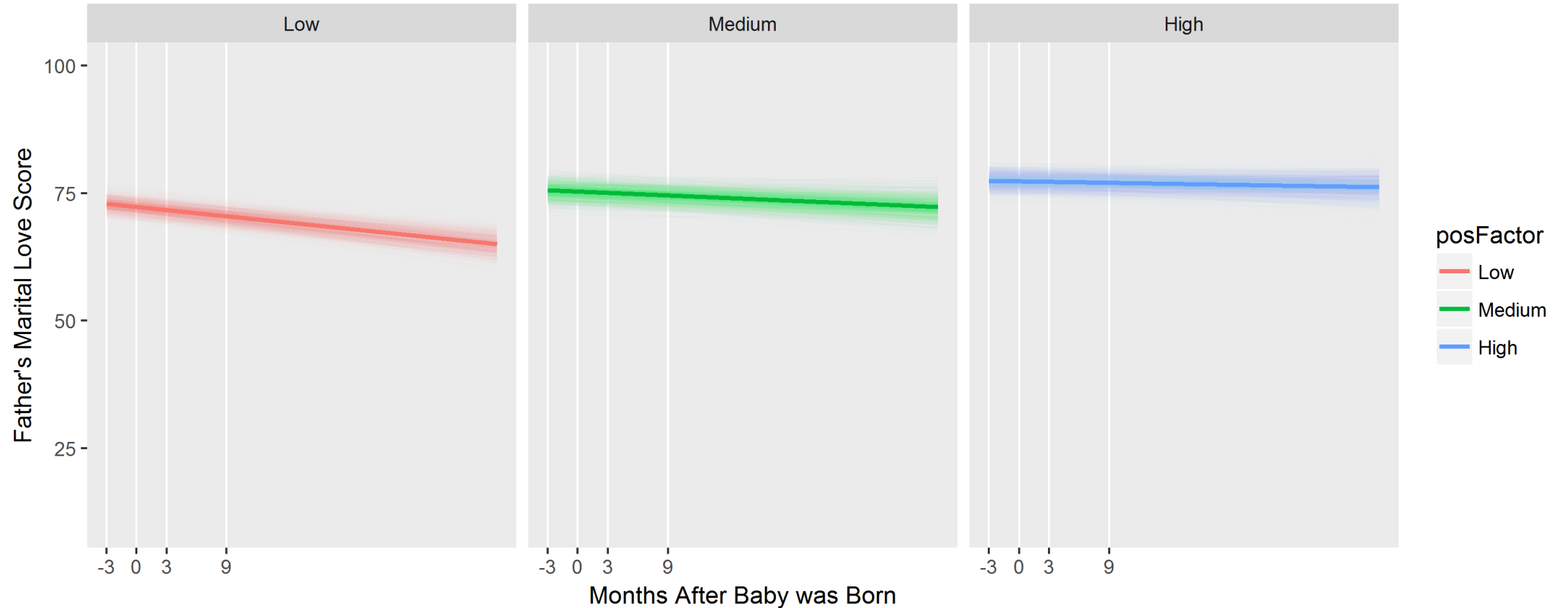
# Individual level models



# Individual level models



# Group level models



# Exploring the posterior distribution

	Mean of posterior distribution	Post. SD = uncertainty around mean (similar to SE)	Post. credibility interval. central 95% of posterior distribution			Highest density interval. 95% range of the distribution with highest probability density		Percentage of post. distr. that is below, in, or above region of practical equivalence (ROPE), here: ROPE = [-0.05,0.05]		
parameter	mean	PSD	PCI	2.50% PCI	97.50% PCI	95% HDI_Low	95% HDI_High	stROPE	inROPE	ltROPE
highPInt	77.363	1.290	74.852	79.881	74.899	79.919	0	0	1	
medPInt	75.266	1.221	72.867	77.640	72.821	77.587	0	0	1	
lowPInt	72.359	1.261	69.873	74.834	69.891	74.850	0	0	1	
highPSlope	-0.032	0.039	-0.110	0.045	-0.108	0.046	0.324	0.657	0.019	
medPSlope	-0.083	0.037	-0.156	-0.010	-0.155	-0.009	0.817	0.182	0.0002	
lowPSlope	-0.203	0.038	-0.280	-0.128	-0.280	-0.128	1	0	0	
betaHighPInt	2.097	1.771	-1.356	5.580	-1.412	5.521	0.111	0.011	0.878	
betaLowPInt	-2.907	1.734	-6.269	0.532	-6.206	0.581	0.947	0.006	0.047	
betaHighPSlope	0.051	0.054	-0.055	0.158	-0.056	0.157	0.031	0.460	0.509	
betaLowPSlope	-0.120	0.053	-0.226	-0.016	-0.226	-0.016	0.909	0.090	0.001	
sdIntercept	6.706	0.616	5.574	7.986	5.531	7.934	0	0	1	
sdSlope	0.120	0.039	0.031	0.187	0.039	0.194	0	0.053	0.947	
corrIntSlope	0.244	0.276	-0.218	0.907	-0.257	0.855	0.122	0.120	0.758	
sdLevel1Error	5.707	0.288	5.172	6.294	5.170	6.290	0	0	1	

# Exploring the posterior distribution

	Mean of posterior distribution	Post. SD = uncertainty around mean (similar to SE)	Post. credibility interval. central 95% of posterior distribution			Highest density interval. 95% range of the distribution with highest probability density		Percentage of post. distr. that is below, in, or above region of practical equivalence (ROPE), here: ROPE = [-0.05,0.05]		
parameter	mean	PSD	PCI	2.50% PCI	97.50% PCI	95% HDI_Low	95% HDI_High	stROPE	inROPE	ltROPE
highPInt	77.363	1.290	74.852	79.881	74.899	79.919	0	0	1	
medPInt	75.266	1.221	72.867	77.640	72.821	77.587	0	0	1	
lowPInt	72.359	1.261	69.873	74.834	69.891	74.850	0	0	1	
highPSlope	-0.032	0.039	-0.110	0.045	-0.108	0.046	0.324	0.657	0.019	
medPSlope	-0.083	0.037	-0.156	-0.010	-0.155	-0.009	0.817	0.182	0.0002	
lowPSlope	-0.203	0.038	-0.280	-0.128	-0.280	-0.128	1	0	0	
betaHighPInt	2.097	1.771	-1.356	5.580	-1.412	5.521	0.111	0.011	0.878	
<div>Entire posterior probability distribution of <b>lowPSlope</b> falls below the ROPE (i.e. below zero) → <b>conclusion: negative slope for low positivity group</b> (i.e. decreasing feelings of marital love over transition into parenthood for fathers who had low positivity at birth of child)  No remarkable upward or downward trend for medium and high groups.</div>										
sdLevel1Error	5.707	0.288	5.172	6.294	5.170	6.290	0	0	1	

Entire posterior probability distribution of **lowPSlope** falls below the ROPE (i.e. below zero)  
**→ conclusion: negative slope for low positivity group** (i.e. decreasing feelings of marital love over transition into parenthood for fathers who had low positivity at birth of child)

No remarkable upward or downward trend for medium and high groups.

# Exploring the posterior distribution

## (planned comparisons)

	Mean of posterior distribution	Post. SD = uncertainty around mean (similar to SE)	Post. credibility interval. central 95% of posterior distribution			Highest density interval. 95% range of the distribution with highest probability density		Percentage of post. distr. that is below, in, or above region of practical equivalence (ROPE), here: ROPE = [-0.05,0.05]		
parameter	mean	PSD	PCI	2.50% PCI	97.50% PCI	95% HDI_Low	95% HDI_High	stROPE	inROPE	ltROPE
c_highLowPInt	5.005	1.814	1.450	8.541	1.539	8.623	0.004	0.001	0.996	
c_highLowPSlope	0.171	0.055	0.063	0.279	0.059	0.274	0.0001	0.013	0.987	
c_highMedPInt	2.068	1.766	-1.417	5.524	-1.506	5.419	0.115	0.012	0.873	
c_highMedPSlope	0.051	0.054	-0.055	0.157	-0.053	0.159	0.031	0.456	0.513	
c_medLowPInt	2.938	1.752	-0.525	6.317	-0.485	6.350	0.045	0.006	0.950	
c_medLowPSlope	0.120	0.053	0.016	0.224	0.016	0.224	0.001	0.094	0.906	

**Contrast High-Low:** 95% HDI above ROPE, ~99% of probability distribution above ROPE.  
→ Conclusion: Compared to fathers in the low positivity group, the high positivity fathers report higher marital love before the birth, and less decline of felt love afterwards.

# Model Comparison

- Deviance information Criterion: How well does the model reduce uncertainty in future predictions?
- Accounts for model complexity and model fit.
- $\text{deviance} = -2 * \log(\text{likelihood})$
- Lower DIC = better model performance in predicting future values
- Compare models in terms of their relative goodness of fit
- >5: some difference,  
>10: considerable difference

**Table 1** Deviance Information Criterion values for five models fitted to the marital love data

Model type	DIC
Quadratic change with positivity grouping	2890
Linear change with positivity grouping	2856
Linear change with no positivity grouping	2855
No change with positivity grouping	2925
No change with no positivity grouping	4905

# Questions

- It takes ~10 minutes to run the model. **What is a good workflow during debugging / modelling to save time?** (smaller data set, less iterations?)
- In MLE the algorithm is always doing the same. Bayesian estimation has all these parameters (chains, burn in, adaptation, iterations, thinning...). **How do I know which values to pick and which are good enough?**
- Related: Despite using the same random seeds, I get minimally different results than the authors. Why?
- Related: The paper has only a brief discussion of model diagnostics (Rhat, convergence checks). What are typical diagnostics. What do I change when there is a problem?
- Related: Instead of DIC other packages advertise other model comparison methods (e.g. WAIC). Which one for what situation?
- Is it ok to just use non-informative priors? **When can I use informative priors, and how do I decide which ones?** → Sensitivity analysis
- This is a lot of code for a standard problem, and getting from the mathematical specification to the model code is not as straightforward as in `lm()` or `lmer()`. **Is there a more concise way to do this?** → packages like `rstanarm`, `brms`, rethinking
- There are different kinds of sampling algorithms (Metropolis, Gibbs, Slice) and different software (Jags, Stan). Why? **Is there a “best choice”?**
- It takes a few seconds to build this model and get the coefficients with `lme4`. The results are almost identical. **Why bother?**
- **Have you used Bayesian data analysis?** What were your experiences with promoters, reviewers...?



# Some more introductory reading

- [Kruschke \(2012\) Bayesian Estimation supersedes the t-test. J Exp Psych.](#)
- [Dienes \(2011\). Bayesian Versus Orthodox Statistics: Which Side Are You On? Perspect Psychol Sc.](#)

## Books:

- Kruschke (2015). Doing Bayesian Data Analysis
- McElrath (2015). Statistical Rethinking

## My code and slides:

<https://github.com/teebusch/jc-bayesgcm>

