Week 7: Item Response Theory

PLSC 40502 - Statistical Models

Review

Previously

- **Cluster models**
  - "Unsupervised" learning of patterns in data
  - Estimation via Expectation-Maximization
- **Topic Modeling**
  - Finite mixture model of text
  - Documents are modeled as mixtures of "topics" and topics are distributions over words
  - Estimation via Variational EM

This week

- **Item Response Theory**
  - Model **discrete** responses as a function of continuous **latent** attributes
  - Logit/probit regression with **unobserved** regressors
- **Ideal point models**
  - Item response theory applied to **legislative voting**
  - Interpreting the latent dimensions

Item Response Theory

Item Response Theory

- In many settings, we observe **binary** or **ordinal** responses to a set of questions among a sample of units.
    - Students' responses to test questions (correcct/incorrect)
    - Expert ratings of countries
    - Legislators voting on bills
- In these settings, we're interested in learning about an (interval scaled) **latent trait** of the units based on their discrete responses
    - Measuring student **ability**
    - Measuring country **characteristics**
    - Measruing legislator **ideology**
- Simple aggregation of the binary responses may give misleading results
    - Some test questions are **harder** than others and provide more information about student ability than questions that everyone gets correct.
    - Averages of ordinal ratings don't have an **interval** interpretation (the distances aren't meaningful)

Item Response Theory

- **Item Response Theory** (IRT) developed out of research on testing and evaluation.
- Observed responses to test questions are a function of:
  - **Latent traits** that are common features of respondents across all questions
  - **Item parameters** that are common features of questions across all respondents.
- **Setup**:
  - Observe N respondents indexed by $i \in \{1, 2, \ldots, N\}$
  - Observe J questions indexed by $j \in \{1, 2, \ldots, J\}$
  - Observe Y_{ij} responses to question i by respondent j
    - We'll work with binary Y_{ij} to start, but can generalize to other outcome distributions (typically ordinal)

One Parameter Logit (1PL)

- The baseline classic IRT model (sometimes called the "Rasch" model after Georg Rasch) assumes the following **item response function**:

$$Pr(Y_{ij} = 1) = F(\theta_i - \alpha_j)$$

where $F()$ is the logistic CDF.

- In other words,

$$Pr(Y_{ij} = 1) = \frac{1}{1 + \exp[-(\theta_i - \alpha_j)]}$$

- Or in "latent variable form"

$$Y_{ij} = \mathbf{1}(Y_{ij}^* > 0)$$

$$Y_{ij}^* = \theta_i - \alpha_j + \epsilon_{ij}$$

where ϵ_{ij} are distributed i.i.d. standard logistic

- Note that alternate $F()$ are common - such as i.i.d. normal ϵ_{ij} which yields the probit version

One Parameter Logit (1PL)

- In the 1PL model, we have **one** parameter describing the item and **one** parameter describing the individual's latent trait

$$Pr(Y_{ij} = 1) = F(\theta_i - \alpha_j)$$

- θ_i is the **latent trait** from unit i
- α_j is the **item difficulty** of question j
- Analogy to the GLM
  - α_j is the **intercept** for task j
  - θ_i is the **regressor** common across all i
  - What are we implicitly assuming?

Two Parameter Logit (2PL)

- One drawback of the 1PL model is that it only allows questions to vary in their difficulty (intercept) and not in the extent to which variation in the responses captures variation in the latent parameters.
  - Essentially assuming a constant "slope" of 1 on the θ parameter
  - But some questions might be bad at capturing θ even if there's variation in Y_{ij}.
- The two-parameter logit adds an additional **item parameter** β_j and assumes

$$Pr(Y_{ij} = 1) = F\Bigg(\beta_j(\theta_i - \alpha_j)\Bigg)$$

- Now each item has:
  - α_j - **item difficulty**
  - β_j - **item discrimination**
- β_j captures the extent to which the question reflects the latent trait
  - β_j close to 0 means that the probability of $Y_{ij} = 1$ is essentially uncorrelated
  - **Negative** β_j implies that **low** latent trait values are more likely to answer $Y_{ij} = 1$ (this creates some identifiability issues!)

Two Parameter Logit (2PL)

- Typically also see the 2PL written as:

$$Pr(Y_{ij} = 1) = F\Bigg(\beta_j\theta_i - \tau_j\Bigg)$$

- Think back again to the logistic regression
  - τ_j is the "intercept"
  - β_j is the "slope"
  - θ_i is the "regressor"

Identification

- While historically, IRT models were estimated via maximum-likelihood, there are many reasons why modern methods use Bayes.

  1. **Inconsistency** - The number of parameters grows as we add more **questions** and as we add more **respondents** so our ML estimators are not consistent
  2. **Non-identifiability** - The 2PL likelihood is invariant to any rescaling of the latent parameters
     - Can multiply all θ by a constant and not change the likelihood
     - Multiplying by -1 changes the interpretation of θ but not the likelihood.

- Putting a prior on θ allows for identification

  - Typically assume $\theta \sim \text{Normal}(0, 1)$
- Also need an additional constraint:
  - Either β is non-negative...
  - ...or one of the θ parameters is fixed to a **known** value

Estimation

- Estimation typically relies on either MCMC or a Variational EM algorithm
    - The underlying **trick** is to use existing theory for Bayesian probit (and now logit) regression to derive the conditional distributions of θ_i, β_j and τ_j
- Consider the "latent variable" form of the logit/probit regression:

$$Y_{ij}^* = \beta_j \theta_i - \tau_j + \epsilon_{ij}$$

- Conditional on θ_i, we have a regression of Y_{ij}^* on θ_i with intercept $-\tau_j$ and slope β_j
- Conditional on β_j and τ_j, we have a regression of $Y_{ij}^* + \tau_j$ on β_j with slope θ_i and no intercept.

Interpretation as a voting model

- In political science, the 2 parameter logit is the standard IRT model for analyzing voting in legislatures
- Often described in terms of **utility maximization**
  - Consider a legislator choosing to vote "Yea" $Y_{ij} = 1$ or "Nay" $Y_{ij} = 0$.
  - These positions are located in some space (we'll work in $\mathbb{R}^1$ for now). The "Yea" position is ζ_j and the "Nay" position is ψ_j.
- Define a utility function for legislator i: $U_i()$
  - $U_i(\zeta_j) = -\frac{1}{\sigma_j}(\theta_i - \zeta_j)^2 + \eta_{ij}$
  - $U_i(\psi_j) = -\frac{1}{\sigma_j}(\theta_i - \psi_j)^2 + \nu_{ij}$
- In other words, they get decreasing utility (in terms of quadratic distance) from policies that are further from their **ideal point**
  - They will vote $Y_{ij} = 1$ if $U_i(\zeta_j) > U_i(\psi_j)$ and $Y_{ij} = 0$ otherwise.
- η_{ij} and ν_{ij} are the "vote-specific error terms"

Interpretation as a voting model

- Returning to the "latent variable" formulation of the logit, we can write: $Y_{ij}^* = U_i(\zeta_j) - U_i(\psi_j)$

- Then, some algebra

$$Y_{ij}^* = -\frac{1}{\sigma_j}(\theta_i - \zeta_j)^2 + \eta_{ij} + \frac{1}{\sigma_j}(\theta_i - \psi_j)^2 - \nu_{ij}$$

$$Y_{ij}^* = \frac{1}{\sigma_j}(-\theta_i^2 + 2\theta_i\zeta_j - \zeta_j^2) + \frac{1}{\sigma_j}(\theta_i^2 - 2\theta_i\psi_j + \psi_j^2) + (\eta_{ij} - \nu_{ij})$$

$$Y_{ij}^* = \frac{2(\zeta_j - \psi_j)}{\sigma_j}\theta_i + \frac{(\psi_j^2 - \zeta_j^2)}{\sigma_j} + (\eta_{ij} - \nu_{ij})$$

- And with some assumptions on the error distribution of $\epsilon_{ij} = (\eta_{ij} - \nu_{ij})$ we have our 2-parameter logit!
  - $\tau_j = -\frac{(\psi_j^2 - \zeta_j^2)}{\sigma_j}$
  - $\beta_j = \frac{2(\zeta_j - \psi_j)}{\sigma_j}$

Interpretation as a voting model

- One implicit assumption in the IRT model is a **conditional independence** assumption in responses given the latent ideology.

$$Pr(Y_{ij} = 1, Y_{ij'} = 1|\theta_i, \beta, \tau) = Pr(Y_{ij} = 1|\theta_i, \beta, \tau) \times Pr(Y_{ij'} = 1|\theta_i, \beta, \tau)$$

- In other words, knowing how a legislator voted on bill j doesn't tell you anything about how they will vote on bill j' if we already know the latent positions of the legislators and bills
  - Could be violated under common legislative behavior (e.g. log-rolling, horse-trading, etc...)
  - You'll still get **something** but interpreting ideal points in this setting is a bit harder - essentially some of the "closeness" in ideal points could be driven by non-ideological factors.

- The "utility maximization" interpretation of the 2PL also implies **single peaked** preferences
  - Legislators strictly prefer policies closer to their ideal point and disprefer ones that are further away
  - But this assumption can be violated if some extreme legislators vote against their party and with the opposition but for differing reasons
    - "Ends against the middle"
    - See **Duck-Mayr and Montgomery (2023)** for a model that tries to account for this.

Example: Improving Polity

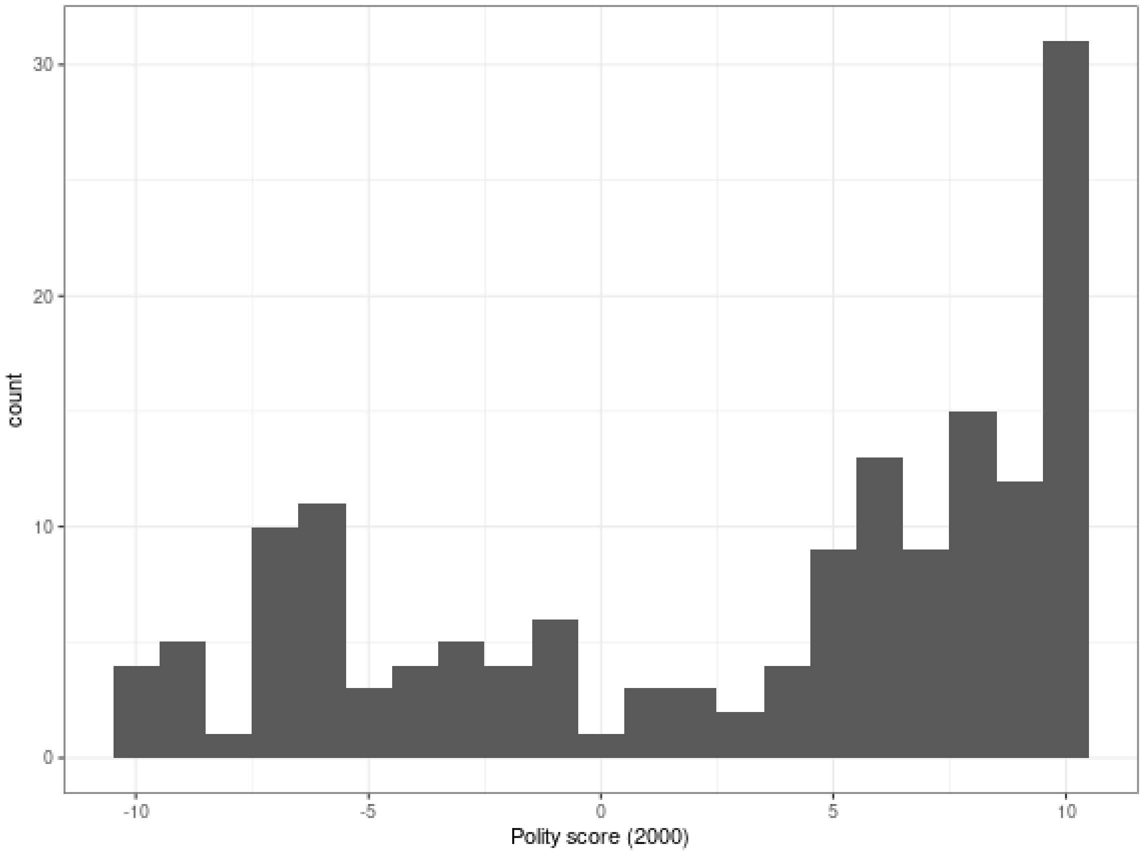
- **Trier and Jackman (2008)** critique the common use of Polity scores as a measure of democracy.
    - Problem of **aggregation** - naively combining the discrete, coded, measures by averaging understates measurement error and may provide a misleading measure of "democratization"
    - Instead propose modeling democracy as a **latent trait** with Polity codings as "expressions" of that latent trait.
- Let's take a look at the most recent Polity data

```
polity <- read_spss("data/p5v2018.sav")
polity2000 <- polity %>% filter(year == 2000) %>% filter(polity!=-88&polity!=-77&polity!=-66)
```

- Polity scores are a **composite** of a set of ordinal measures related to
    1. **Executive recruitment**
    2. **Executive constraint**
    3. **Political competition**

Example: Improving Polity

- The standard approach is to **aggregate** these scores into a "democracy" and an "autocracy" index that are added together to yield a score from -10 to 10



Example: Improving Polity

- The component scores are often not independent of one another
  - For example, "xropen" captures the extent to which executive elections are "open" but it's partially constrained by "xrcomp", the competitiveness of executive recruitment.

```
table(polity2000$xropen, polity2000$xrcomp)
```

```
##
##       0  1  2  3
##   0 24  0  0  0
##   1  0  6  0  0
##   2  0  5  0  0
##   3  0  0  2  0
##   4  0 20 33 65
```

Example: Improving Polity

- **Trier and Jackman (2008)** settle on three ordinal indices constructed from the Polity components as the latent "tasks" for their IRT model

```
table(polity2000$exrec)
```

```
## 
##  1  2  3  4  5  6  7  8 
##  6  5 20 14 10  2 33 65
```

```
table(polity2000$exconst)
```

```
## 
##  1  2  3  4  5  6  7 
## 15 13 26  8 25 18 50
```

```
table(polity2000$polcomp)
```

```
## 
##  1  2  3  4  5  6  7  8  9 10 
## 16 19  7  1  1 18 15 10 35 33
```

- Each of these ordinal indicators is modeled as an expression of some underlying latent "democracy" variable

Example: Improving Polity

- The model used in the paper is a **two-parameter** ordinal logit.
  - Index country-year by i, Polity indicator by j, and the K_j ordinal categories by k.
  - Latent "democracy" variable θ_i and latent "item discrimination parameter" β_j
- The probability of observing a rating for country i on indicator j, Y_{ij} is:

$$\begin{aligned} Pr(Y_{ij} = 1) &= F(\tau_{j1} - \theta_i\beta_j) \\ &\vdots \\ Pr(Y_{ij} = k) &= F(\tau_{jk} - \theta_i\beta_j) - F(\tau_{j,k-1} - \theta_i\beta_j) \\ &\vdots \\ Pr(Y_{ij} = K_j) &= 1 - F(\tau_{j,K_j-1} - \theta_i\beta_j) \end{aligned}$$

where $F()$ is the logistic CDF and $\tau_{j1} < \tau_{j2} < \ldots < \tau_{j,K_j-2} < \tau_{j,K_j-1}$ are a set of ordered cut-points for indicator j

Example: Improving Polity

- Let's implement this for 2000 in Stan (we could do a full model for the entire Polity dataset, but that takes longer to run)
- First, the data block

```
data{
  int<lower=1> N; // number of countries
  int<lower=1> K[3]; // number of categories per task
  int<lower=1> Y[N,3]; // responses to each task (hard-coding 3 tasks)
}
```

Example: Improving Polity

- For the parameters, we'll define θ and β as usual...
  - ...but we'll use a trick to re-parameterize the cutpoints to make it easier to put priors on them!

```
parameters{
array[N] real theta; // country scores;
array[3] real beta; // discrimination parameters;
array[3] real delta_start; // parameterization of cutpoints;
vector<lower=0>[K[1]-2] delta1; // parameterization of distances
vector<lower=0>[K[2]-2] delta2;
vector<lower=0>[K[3]-2] delta3;
}
```

Example: Improving Polity

- The cutpoints are **ordered**
    - But we want to put a prior on an **unordered** parameters
    - **Solution**: Put a normal prior on the first cutpoint and then the **distances** between each gap!
    - Use the `transformed parameters` block to generate the ordered `tau`

```
transformed parameters{
  ordered[K[1]-1] tau1;
  tau1[1] = delta_start[1];
  tau1[2:] = delta_start[1] + cumulative_sum(delta1);
  ordered[K[2]-1] tau2;
  tau2[1] = delta_start[2];
  tau2[2:] = delta_start[2] + cumulative_sum(delta2);
  ordered[K[3]-1] tau3;
  tau3[1] = delta_start[3];
  tau3[2:] = delta_start[3] + cumulative_sum(delta3);
}
```

Example: Improving Polity

- Lastly our model using the ordered logistic specification

```
model{
  theta ~ normal(0, 1);
  beta ~ normal(0, 9);
  delta_start ~ normal(0, 6.6666667);
  delta1 ~ exponential(2);
  delta2 ~ exponential(2);
  delta3 ~ exponential(2);
  for (n in 1:N) {
    Y[n,1] ~ ordered_logistic(theta[n] * beta[1], tau1);
    Y[n,2] ~ ordered_logistic(theta[n] * beta[2], tau2);
    Y[n,3] ~ ordered_logistic(theta[n] * beta[3], tau3);
  }
}
```

Example: Improving Polity

- Pass in the data

```
polity_data <- list(N = nrow(polity2000), K= apply(polity2000 %>% select(exrec, exconst,polcomp
                  Y= polity2000 %>% select(exrec, exconst,polcomp))
```

- Estimate the model (run 4 chains)
  - **Warning**: This model actually has convergence problems across the different chains

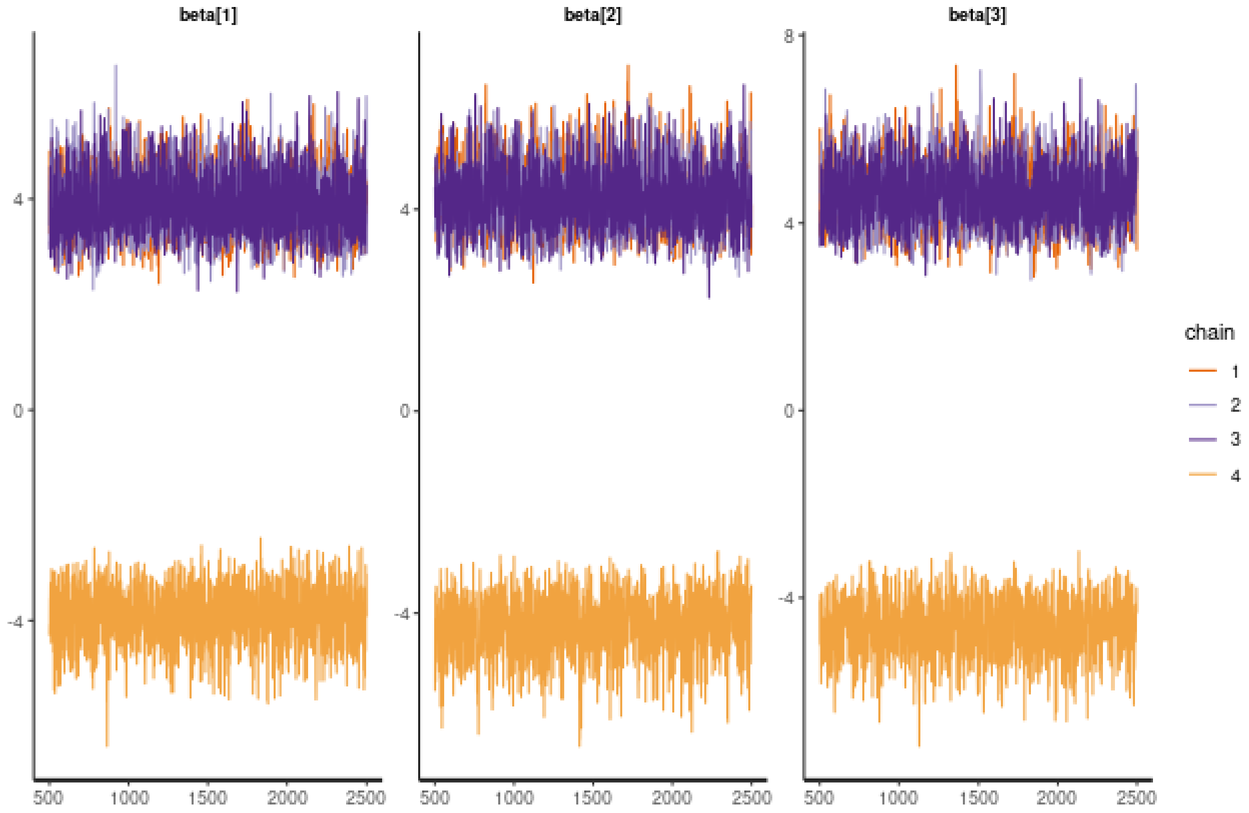
```
# This model has convergence problems
polity_irt_bad <-  stan(
  model_code = polityirt_model,  # Stan code
  data = polity_data,     # named list of data
  chains = 4,              # number of Markov chains
  warmup = 500,           # number of warmup iterations per chain
  iter = 2500,             # total number of iterations per chain
  cores = 4,               # number of cores (could use one per chain - by default uses however
  refresh = 0,
  seed = 60637
  )
```

```
## Running /usr/lib/R/bin/R CMD SHLIB foo.c
## gcc -I"/usr/share/R/include" -DNDEBUG   -I"/home/anton/R/x86_64-pc-linux-gnu-library/4.2/Rcpp/include
## In file included from /home/anton/R/x86_64-pc-linux-gnu-library/4.2/RcppEigen/include/Eigen/Core:88,
```

Example: Improving Polity

- What's happening here?

```
traceplot(polity_irt_bad, pars=c("beta"))
```



Example: Improving Polity

Because the IRT model is identified only up to scaling, the MCMC ends up going to one of two posterior modes

- β is positive and high θ reflects high democracy
- or β is negative and high θ reflects high **autocracy**

Example: Improving Polity

- Essentially, we need to put some additional constraints on the model.
- One easy fix for this version is to make the discrimination parameters **always** positive
    - This is standard for the education/testing IRT model
    - We wouldn't want this for a voting model, but for the democracy model it's not unreasonable
    - Instead of a normal prior on β, we can use a log-normal or half-normal.
- In a voting model, we want to allow for votes to have both a positive and a negative discrimination parameter
    - Some votes have Democrats voting **Yes** and Republicans voting **No**; others have Democrats voting **No** and Republicans voting **Yes**
    - We'll instead fix some known legislators' latent ideal points to particular values using a "spike" prior

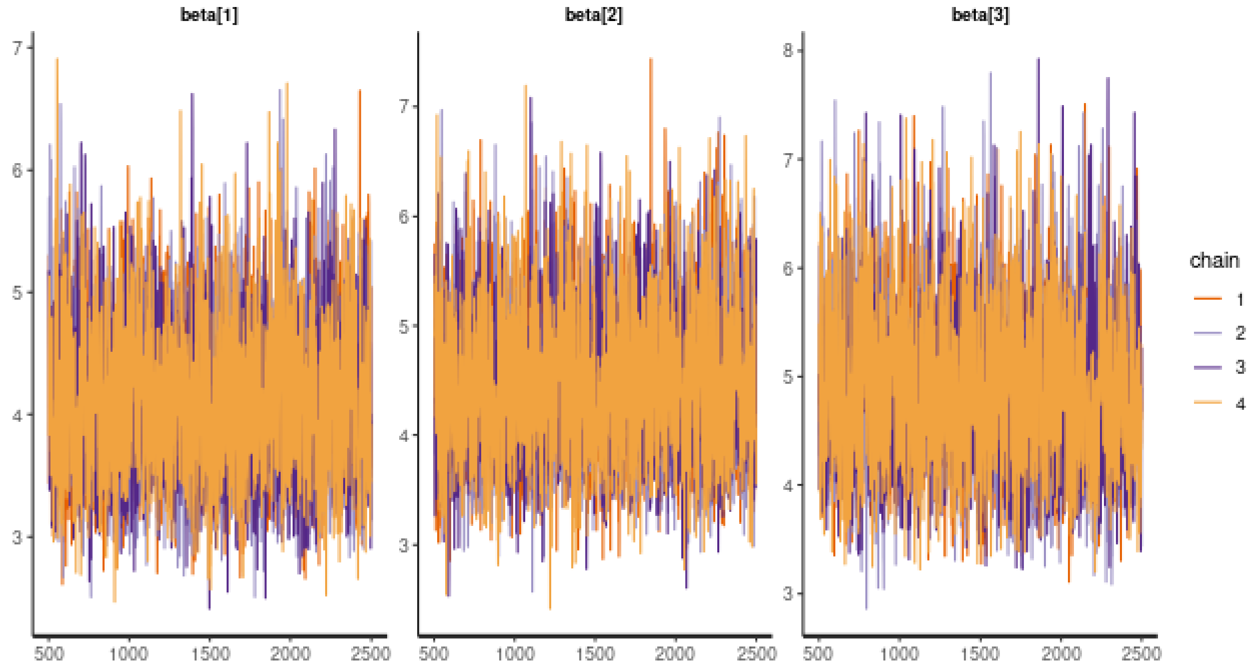
Example: Improving Polity

```
# In this model, we restrict the betas to be positive
polity_irt <-  stan(
  model_code = polityirt_model_fixed,  # Stan code
  data = polity_data,    # named list of data
  chains = 4,             # number of Markov chains
  warmup = 500,          # number of warmup iterations per chain
  iter = 2500,            # total number of iterations per chain
  cores = 4,              # number of cores (could use one per chain - by default uses however
  refresh = 0,
  seed = 60637
  )
```

Example: Improving Polity

- Problem solved!

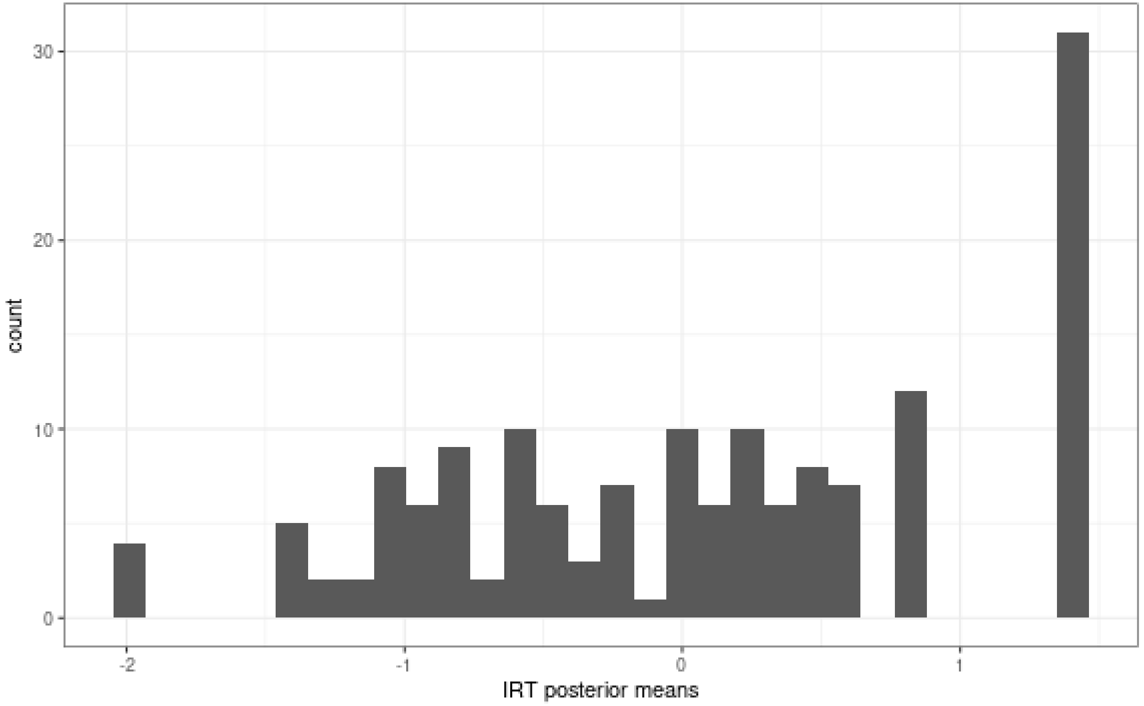
```
traceplot(polity_irt, pars=c("beta"))
```



Example: Improving Polity

- Get the latent democracy scores

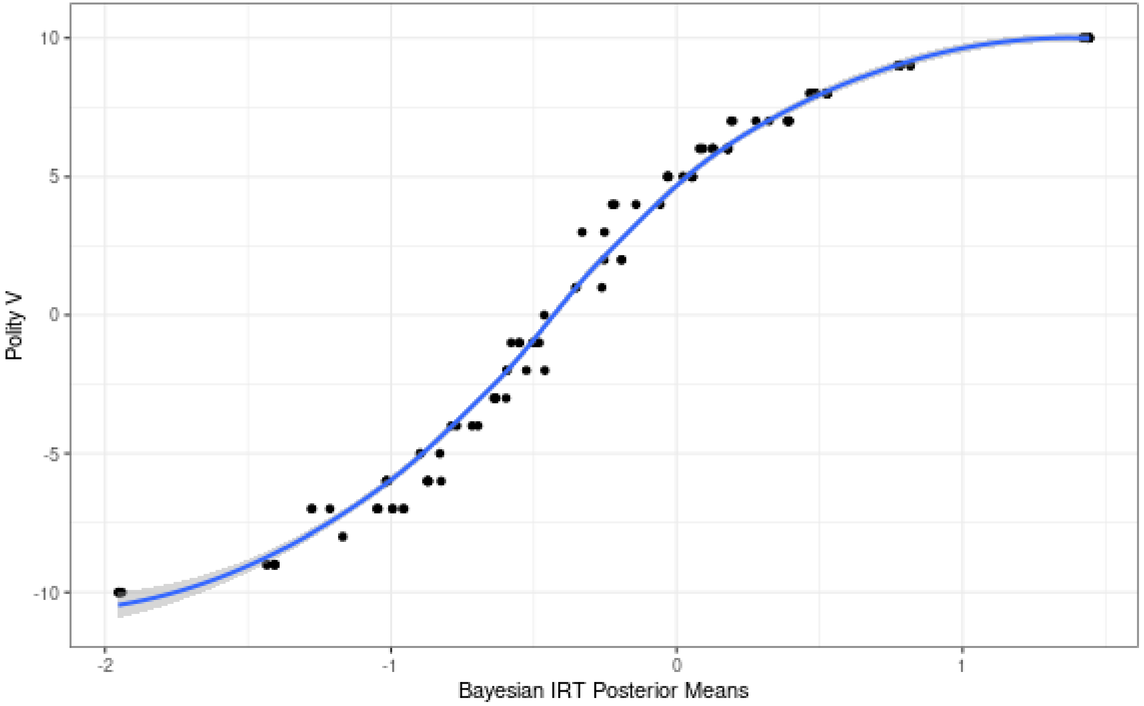
```
democracy_scores <- rstan::extract(polity_irt)$theta
polity2000$pm_irt <- colMeans(democracy_scores)
polity2000 %>% ggplot(aes(x=pm_irt)) + geom_histogram() + xlab("IRT posterior means") + theme_b
```



Example: Improving Polity

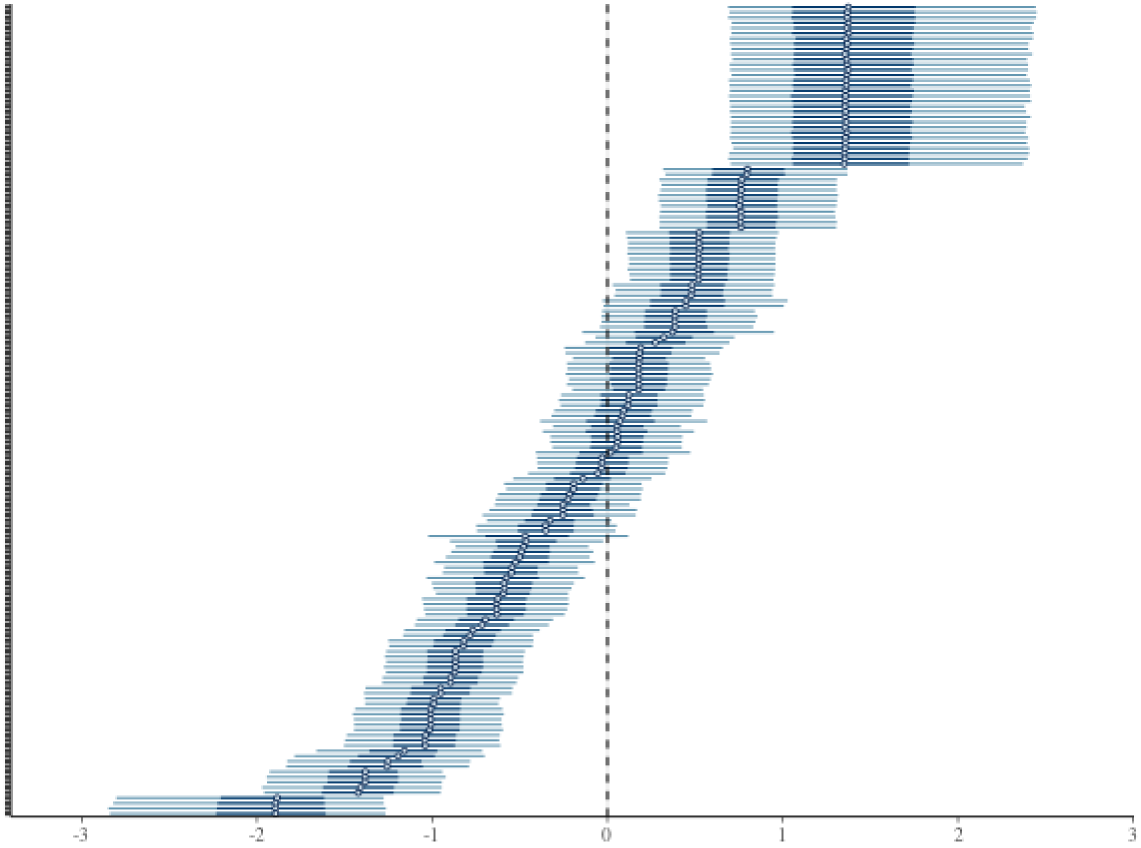
- Plot against the "Polity" scores

```
polity2000 %>% ggplot(aes(x=pm_irt, y=polity2)) + geom_point() + geom_smooth() + xlab("Bayesiar
```



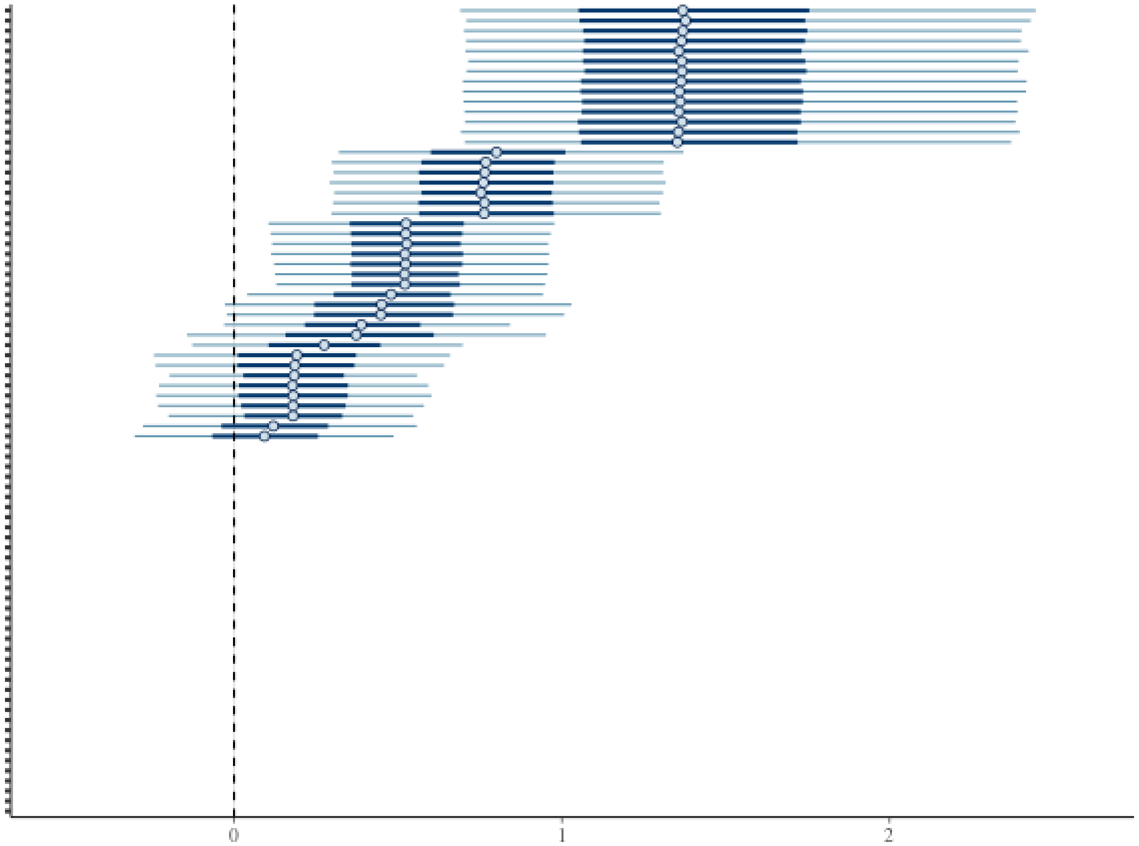
Example: Improving Polity

- Plot posteriors by country



Example: Improving Polity

- A common classification of states as "democracies" that is commonly used is $\mathbf{Polity} \geq 6$
  - What does that look like for the posteriors of those states?



Example: Improving Polity

- How about the "non-democracies" $\textbf{Polity} < 6$?