Models with Ordinal Variables Using the brms R Package

Ben Goodrich

April 18, 2022

Distributions of Different Random Variables

- α and each β_k have a posterior (or prior) distribution in a regression model
- Let $\eta_n = \alpha + \sum_{k=1}^{K} \beta_k x_{nk}$. The `posterior_linpred` function produces draws of each η_n induced by the posterior distribution of α and each β_k
- In a GLM, $\mu_n = g\left(\eta_n\right)$. The `posterior_epred` function produces draws of each μ_n induced by the posterior distribution of η_n
- The P{D,M}F of the outcome is $f\left(y_n \mid \mu_n, \dots\right)$. The `posterior_predict` function produces draws of each y_n induced by the posterior distribution of μ_n whose P{D,M}F is $f\left(y_n \mid \mu_n, \dots\right)$
- But y_n is not conditionally deterministic given μ_n because it includes noise, whose posterior distribution may be governed by other parameters like σ
- In the case of a logit model, $\eta_n \in \mathbb{R}$, $\mu_n = \frac{1}{1+e^{-\eta_n}} \in (0,1)$, and $y_n \in \{0,1\}$

Censored Observations (with a spline)

```r
data(kidney, package = "brms")
head(kidney)
```

```
##   time censored patient recur age    sex disease
## 1    8        0       1     1  28   male   other
## 2   23        0       2     1  48 female      GN
## 3   22        0       3     1  32   male   other
## 4  447        0       4     1  31 female   other
## 5   30        0       5     1  10   male   other
## 6   24        0       6     1  16 female   other
```

```r
prior <- brm(time | cens(censored) ~ s(age, by = sex) + disease,
             data = kidney, family = lognormal(), sample_prior = "only",
             prior = prior(normal(0, 2), class = "b") +
               prior(normal(-15, 3), class = "Intercept") +
               prior(exponential(0.1), class = "sigma"))
```

Checking the Prior Predictive Distribution

```
prior_PD <- posterior_predict(prior)
dim(prior_PD)
```

```
## [1] 4000   76
```

```
summary(colMeans(prior_PD))
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 3.963e+27 1.523e+41 8.855e+48 8.770e+85 2.393e+60 6.635e+87
```

- This is terrible but happens a lot when researchers increase the complexity of their models without increasing the amount of effort they put into choosing good priors on the parameters

Results of the Right Censored Model

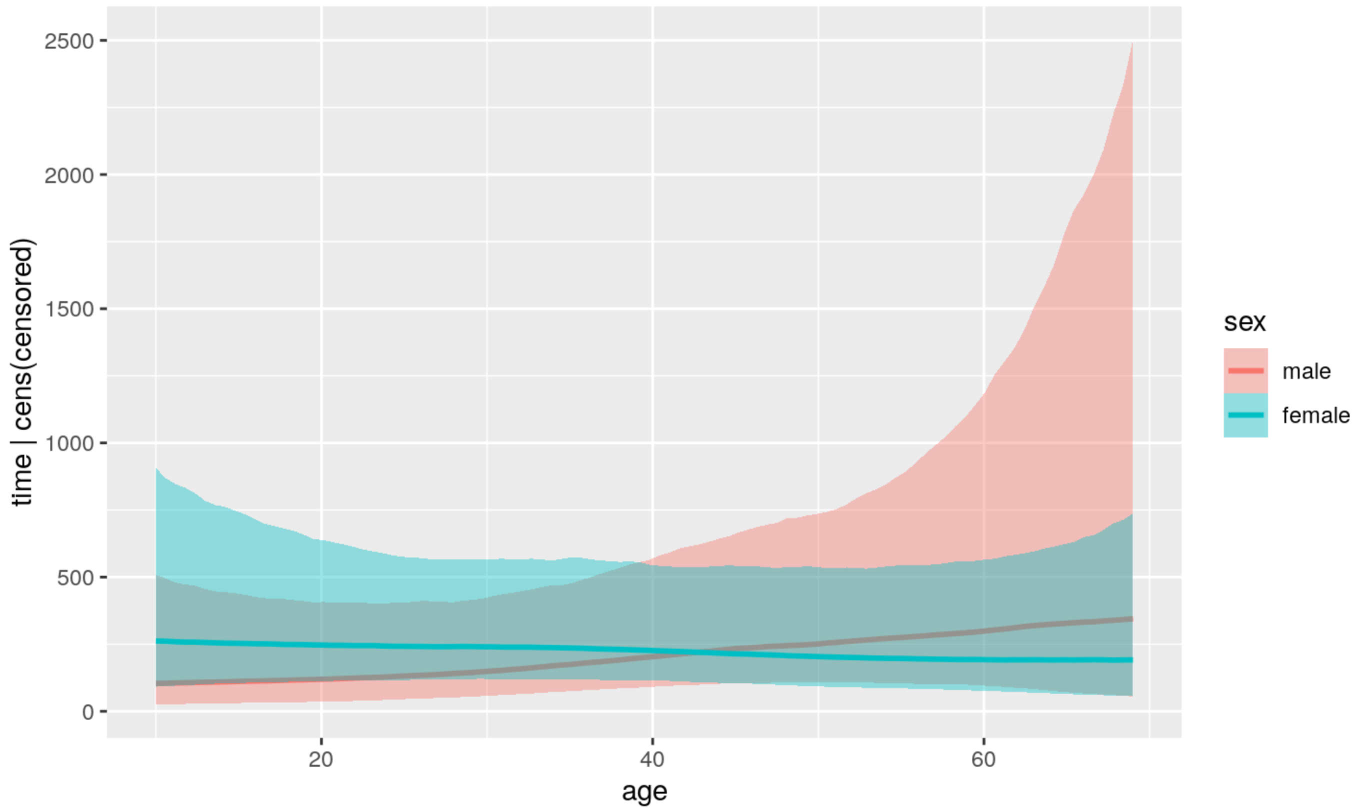
```
post <- update(prior, sample_prior = "no", control = list(adapt_delta = 0.99))
```

```
post
```

```
...
## Smooth Terms:
##                        Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sds(sagesexmale_1)         1.10      1.25     0.03     3.98 1.00     2047     1843
## sds(sagesexfemale_1)       0.76      0.77     0.02     2.82 1.00     1855     1442
##
## Population-Level Effects:
##                    Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept              4.45      0.33     3.81     5.10 1.00     2527     2772
## diseaseGN             -0.62      0.50    -1.60     0.37 1.00     3176     3131
## diseaseAN             -0.39      0.48    -1.33     0.57 1.00     2938     2842
## diseasePKD             0.16      0.65    -1.14     1.44 1.00     3644     3391
## sage:sexmale_1         0.77      1.66    -2.57     3.86 1.00     3924     3301
## sage:sexfemale_1      -0.23      1.36    -2.83     2.59 1.00     2703     2230
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     1.37      0.14     1.13     1.68 1.00     4737     3186
##
```

Plot of μ_n versus age_n

```r
plot(conditional_effects(post, effects = "age:sex"))
```



Warnings You Should Be Aware Of

1. Divergent Transitions: This means the tuned stepsize ended up too big relative to the curvature of the log-kernel. Increase `adapt_delta` above its default value (usually 0.8) and / or use more informative priors
2. Hitting the maximum treedepth: This means the tuned stepsize ended up so small that it could not get all the way around the parameter space in one iteration. Increase `max_treedepth` beyond its default value of 10 but each increment will double the wall time, so only do so if you hit the max a lot
3. Bulk / Tail Effective Sample Size too low: This means the tuned stepsize ended up so small that adjacent draws have too much dependence. Increase the number of iterations or chains
4. $\widehat{R} > 1.01$: This means the chains have not converged. You could try running the chains longer, but there is probably a deeper problem.
5. Low Bayesian Fraction of Information: This means that you posterior distribution has really extreme tails. You could try running the chains longer, but there is probably a deeper problem.

Data-Generating Process for Interval Outcomes

$$\alpha \sim ???$$
$$\forall k : \beta_k \sim ???$$
$$\forall n : \mu_n \equiv \alpha + \sum_{k=1}^{K} \beta_k x_{nk}$$
$$\sigma \sim ???$$
$$\forall n : \epsilon_n \sim \mathcal{N}\left(0, \sigma\right)$$
$$\forall n : y_n^* \equiv \mu_n + \epsilon_n$$
$$y_n \equiv \sum_{j=1}^{J-1} \mathbb{I}\{y_n^* > z_j\}$$

Each z_j is a KNOWN cutpoint, such as in “Is your family income between \$0 and \$20,000, \$20,000 and \$50,000, \$50,000 and \$100,000, \$100,000 and \$200,000, or more than \$200,000?”

Log-Likelihood for Interval Outcomes

$$\ell\left(\alpha, \beta_1, \dots, \beta_K, \sigma\right) = \sum_{n=1}^{N} \ln \Pr\left(y_n \mid \alpha, \beta_1, \dots, \beta_K, \sigma\right) =$$
$$\sum_{n=1}^{N} \ln\Big(F\left(z_{y_n} \mid \mu_n, \sigma\right) - F\left(z_{y_n - 1} \mid \mu_n, \sigma\right)\Big)$$

where F is the normal CDF (but could easily be another CDF).

```
brm(z[y - 1]  | cens("interval", z[y]) ~ x1 + ... xk,
    data = dataset, family = gaussian, prior = ???)
```

Data-Generating Process for Ordinal Outcomes

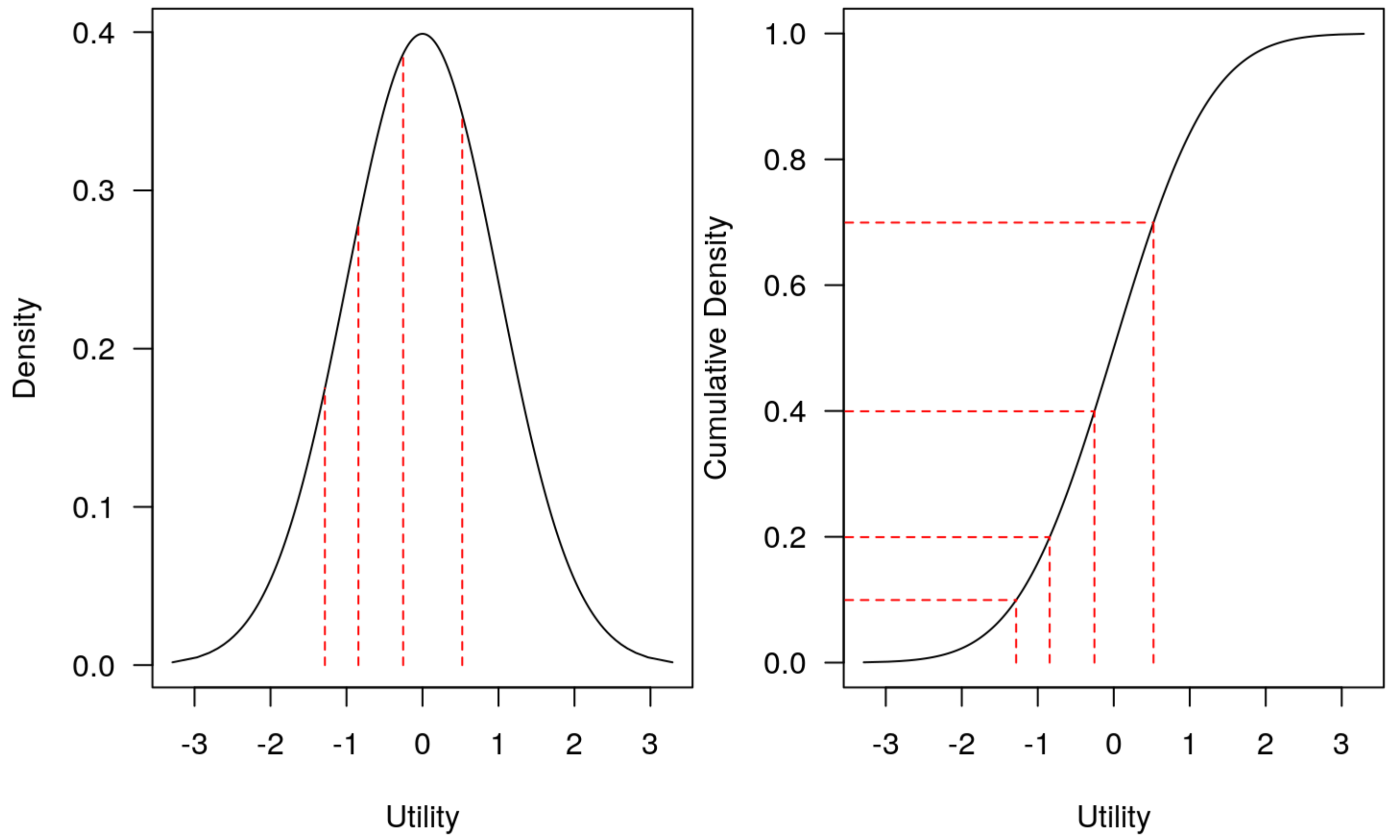
$$\forall k : \beta_k \sim ???$$
$$\forall n : \eta_n \equiv \sum_{k=1}^{K} \beta_k x_{nk}$$
$$\forall n : \epsilon_n \sim \mathcal{N}(0,1)$$
$$\forall n : y_n^* \equiv \eta_n + \epsilon_n$$
$$\zeta_1 \equiv -\infty$$
$$\forall j > 1 : \zeta_j \sim ???$$
$$y_n \equiv \sum_{j=1}^{J-1} \mathbb{I}\{y_n^* > \zeta_j\}$$

- Each ζ_j is a UNKNOWN cutpoint (if $j > 1$), such as in “Do you approve, neither approve nor disapprove, or disapprove of the job Joe Biden is doing as President?” to estimate
- $\alpha \equiv 0$ because you could shift α by any constant & shift each ζ_j by the same constant without affecting y_n
- $\sigma \equiv 1$ because you could scale each y_n^* by any positive constant & scale each ζ_j by the same constant without affecting y_n, i.e. only RELATIVE values of y_n^* matter

Likelihood for an Ordered Observation

- Likelihood for an observation is just categorical: $\mathcal{L}(\beta, \boldsymbol{\zeta}; y) \propto \prod_{j=1}^{J} \Pr(y = j | \beta, \boldsymbol{\zeta})$
- If $F()$ is in the location-scale family (normal, logistic, etc.), then $F(\beta x + \epsilon \leq \zeta_j) = F_{0,1}(\zeta_j - \beta x)$, where $F_{0,1}()$ is the "standard" version of the CDF
- $\Pr(y = j | \beta, \boldsymbol{\zeta}) = F(\beta x + \epsilon \leq \zeta_j) - F(\beta x + \epsilon \leq \zeta_{j-1})$
- Bernoulli is a special case with only two categories

Graphs of Standard Normal Utility with Cutpoints



Estimating an Ordinal Model with `stan_polr`

```
library(rstanarm); options(mc.cores = parallel::detectCores())
data("inhaler", package = "brms")
inhaler$rating <- as.ordered(inhaler$rating)
post <- stan_polr(rating ~ treat + period + carry, data = inhaler,
                  method = "probit", prior = R2(0.25), seed = 12345)
```

- Now we can estimate the causal effect of `treat` on utility for `rating`:

```
nd <- inhaler; nd$treat <- 1
y1_star <- posterior_linpred(post, newdata = nd)
nd$treat <- 0
y0_star <- posterior_linpred(post, newdata = nd)
summary(c(y1_star - y0_star))
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.99800 -0.58823 -0.49167 -0.49074 -0.39224  0.06241
```

Results of `rstanarm::stan_polr`

```
print(post, digits = 2)
```

```
...
## ------
##        Median MAD_SD
## treat  -0.49   0.15
## period  0.11   0.10
## carry  -0.12   0.10
##
## Cutpoints:
##     Median MAD_SD
## 1|2 0.33   0.05
## 2|3 1.77   0.09
## 3|4 2.27   0.14
##
...
```

Dirichlet Distribution

- Dirichlet distribution is over the parameter space of PMFs — i.e. $\pi_k \geq 0$ and $\sum_{k=1}^{K} \pi_k = 1$ — and the Dirichlet PDF is $f\left(\boldsymbol{\pi} \mid \boldsymbol{\alpha}\right) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{k=1}^{K} \pi_k^{\alpha_k - 1}$ where $\alpha_k \geq 0 \, \forall k$ and the multivariate Beta function is $B\left(\boldsymbol{\alpha}\right) = \frac{\prod_{k=1}^{K} \Gamma(\alpha_k)}{\Gamma\left(\prod_{k=1}^{K} \alpha_k\right)}$ where $\Gamma\left(z\right) = \frac{1}{z} \prod_{n=1}^{\infty} \frac{\left(1+\frac{1}{n}\right)^n}{1+\frac{z}{n}} = \int_0^{\infty} u^{z-1} e^{-u} du$ is the Gamma function
- $\mathbb{E}\pi_i = \frac{\alpha_i}{\sum_{k=1}^{K} \alpha_k} \forall i$ and the mode of π_i is $\frac{\alpha_i - 1}{-1 + \sum_{k=1}^{K} \alpha_k}$ if $\alpha_i > 1$
- Iff $\alpha_k = 1 \, \forall k$, $f\left(\boldsymbol{\pi} \middle| \, \boldsymbol{\alpha} = \mathbf{1}\right)$ is constant over Θ (simplexes)
- Beta distribution is a special case of the Dirichlet where $K = 2$
- Marginal and conditional distributions for subsets of $\boldsymbol{\pi}$ are also Dirichlet

Priors on Cutpoints

- `stan_polr` puts a Dirichlet prior (by default, with $\alpha_k = 1 \forall k$) on the probability a unit with average predictors would have y_k as its outcome
- The cutpoints, $\boldsymbol{\zeta}$, are derived from this by inverting the inverse link function. In R, it would look like

```
simplex <- MCMCpack::rdirichlet(n = 1, alpha = rep(1, 5)); rbind(simplex, cumsum(simplex))
```

```
##           [,1]      [,2]      [,3]      [,4]      [,5]
## [1,] 0.1514626 0.2081140 0.1607980 0.1003763 0.3792491
## [2,] 0.1514626 0.3595766 0.5203746 0.6207509 1.0000000
```

```
(zeta <- qnorm(cumsum(simplex)))
```

```
## [1] -1.03018051 -0.35959070  0.05109374  0.30745348         Inf
```

- However, `brms::brm` does something quite different, by default

Similar Model with `brms::brm`

- `brm` can estimate similar models, but with priors on the coefficients

```
post <- brm(rating ~ treat + period + carry, data = inhaler,
            family = cumulative(link = "probit"),
            prior = prior("logistic(0, 1)", class = "b"))
```

```
post # Intercept[j] corresponds to cutpoint[j] from stan_polr
```

```
...
##
## Population-Level Effects:
##              Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept[1]     0.34      0.05     0.23     0.44 1.00     4807     3282
## Intercept[2]     1.79      0.09     1.61     1.98 1.00     5716     3249
## Intercept[3]     2.32      0.14     2.04     2.61 1.00     5601     2924
## treat           -0.49      0.15    -0.78    -0.20 1.00     3599     2822
## period           0.12      0.10    -0.08     0.33 1.00     5209     3012
## carry           -0.12      0.10    -0.33     0.08 1.00     3673     3099
##
## Family Specific Parameters:
##      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## disc     1.00      0.00     1.00     1.00 1.00     4000     4000
##
```

Can use `loo` (if you had multiple models)

```
loo(post)
```

```
##
## Computed from 4000 by 572 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo   -458.4 17.1
## p_loo         6.0  0.6
## looic       916.9 34.3
## ------
## Monte Carlo SE of elpd_loo is 0.0.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```

Data-Generating Process with Ordinal Predictors

$$\alpha \sim ???$$
$$\forall k : \beta_k \sim ???$$
$$\theta_1, \dots, \theta_{J-1} \sim Dir\left(a_1, \dots, a_{J-1}\right)$$
$$\gamma \sim ???$$
$$\forall n : \mu_n \equiv \alpha + \sum_{k=1}^{K} \beta_k x_{nk} +$$
$$J\gamma \sum_{j=1}^{c_n - 1} \theta_j$$
$$\sigma \sim ???$$
$$\forall n : \epsilon_n \sim \mathcal{N}\left(0, \sigma\right)$$
$$\forall n : y_n \equiv \mu_n + \epsilon_n$$

- Each c_n is a KNOWN category, such as in "Is your family income between \$0 and \$20,000, \$20,000 and \$50,000, \$50,000 and \$100,000, \$100,000 and \$200,000, or more than \$200,000?"
- γ can be interpreted as the average effect of going up one more category
- Since $0 \leq \sum_{j=1}^{c_n - 1} \theta_j \leq 1$, the sum is the fraction of $J\gamma$ of going from lowest category to c_n

Ordinal Predictors in Polling

```
poll <- readRDS("GooglePoll.rds") # WantToWin is coded as 1 for Romney and 0 for Obama
library(dplyr)
collapsed <- filter(poll, !is.na(WantToWin)) %>%
             group_by(Region, Gender, Urban_Density, Age, Income) %>%
             summarize(Romney = sum(grepl("Romney", WantToWin)), Obama = n() - Romney) %>%
             na.omit

post <- brm(Romney | trials(Romney + Obama) ~ Region + Gender + Urban_Density +
              # Age and Income are restricted to have monotonic effects
              mo(Age) + mo(Income), data = collapsed, family = binomial(link = "logit"),
            prior = prior("logistic(0,1)", class = "b"))
```

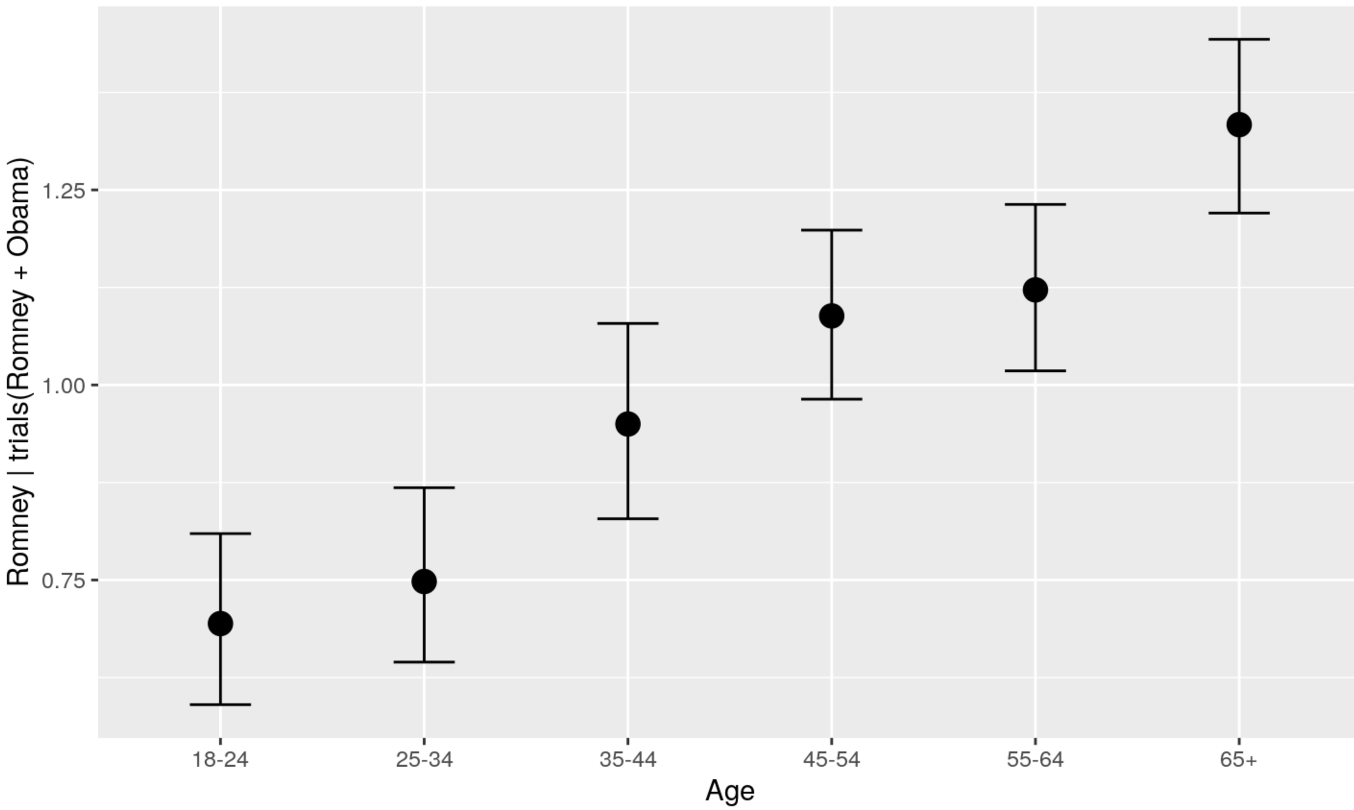
- For more examples, see https://cran.r-project.org/package=brms/vignettes/brms_monotonic.html

Results of Model with Ordinal Predictors

```
...
##                        Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept                 -0.63      0.12    -0.87    -0.39 1.00     2984     2688
## RegionNORTHEAST           -0.13      0.09    -0.32     0.05 1.00     3580     3066
## RegionSOUTH                0.31      0.07     0.17     0.45 1.00     3152     2957
## RegionWEST                -0.14      0.08    -0.29     0.01 1.00     3109     3122
## GenderMale                 0.39      0.06     0.28     0.50 1.00     4592     2911
## Urban_DensitySuburban     -0.19      0.09    -0.36    -0.01 1.00     2867     2734
## Urban_DensityUrban        -0.50      0.09    -0.67    -0.32 1.00     2885     2620
## moAge                      0.27      0.02     0.23     0.30 1.00     3130     2936
## moIncome                   0.01      0.06    -0.09     0.14 1.00     2199     1953
##
## Simplex Parameters:
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## moAge1[1]         0.09      0.05     0.00     0.21 1.00     2835     1554
## moAge1[2]         0.31      0.07     0.18     0.46 1.00     3594     2721
## moAge1[3]         0.21      0.07     0.08     0.34 1.00     3092     1989
## moAge1[4]         0.05      0.04     0.00     0.14 1.00     3474     1935
## moAge1[5]         0.34      0.06     0.22     0.44 1.00     5742     3761
## moIncome1[1]      0.18      0.16     0.01     0.59 1.00     3098     2098
## moIncome1[2]      0.15      0.14     0.00     0.52 1.00     3590     2227
## moIncome1[3]      0.19      0.16     0.01     0.59 1.00     4020     2290
## moIncome1[4]      0.23      0.18     0.01     0.67 1.00     3737     2454
## moIncome1[5]      0.25      0.19     0.01     0.70 1.00     3096     2585
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
...
```

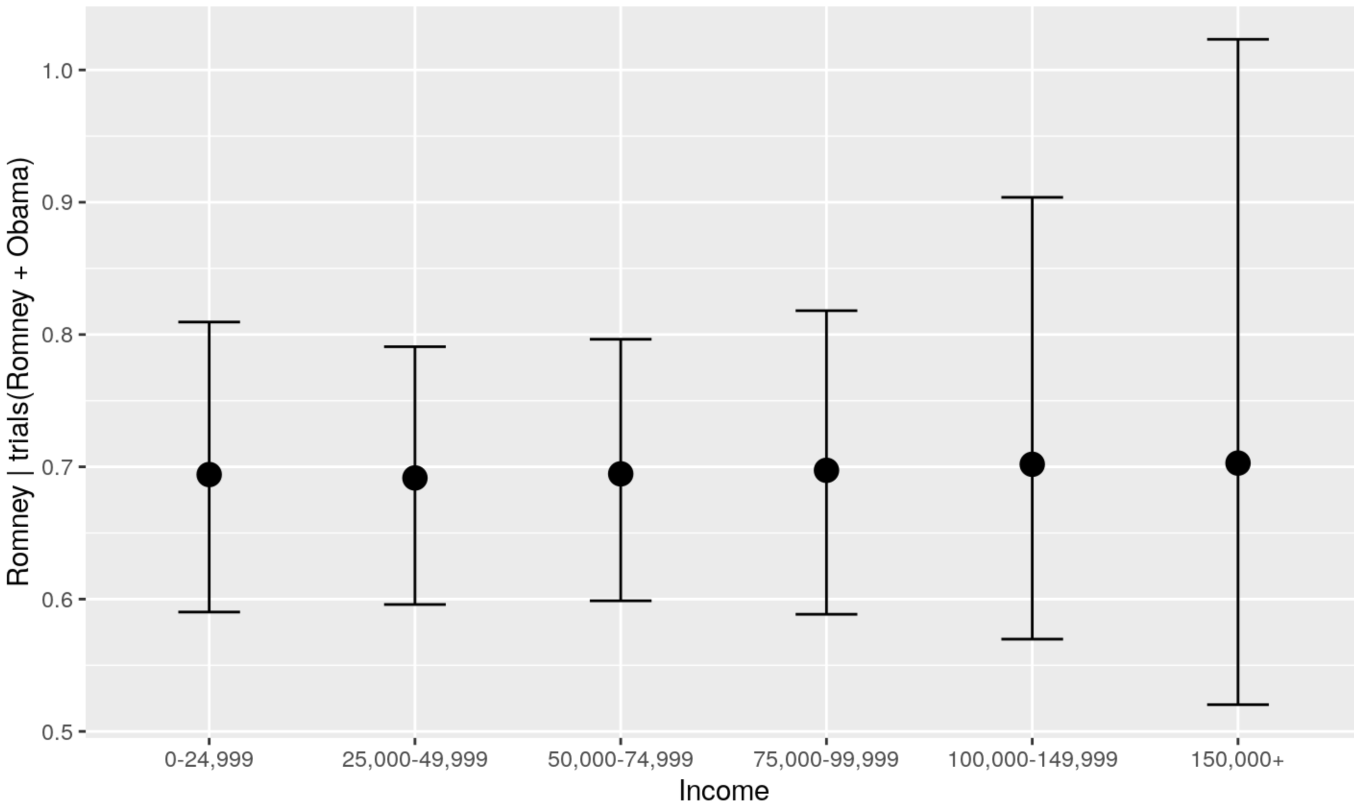
Effect of Age Plot

```r
plot(conditional_effects(post, effects = "Age")) # vertical axis is in log-odds
```



Effect of Income Plot

```
plot(conditional_effects(post, effects = "Income")) # forced monotonic but maybe wrong?
```



Try It without the Restriction on Income

```
post2 <- brm(Romney | trials(Romney + Obama) ~ Region + Gender + Urban_Density +
              mo(Age) + Income, data = collapsed, family = binomial(link = "logit"),
            prior = prior("logistic(0,1)", class = "b"))
```

```
post <- add_criterion(post, criterion = "loo")
post2 <- add_criterion(post2, criterion = "loo")
loo_compare(post, post2)
```

```
##       elpd_diff se_diff
## post   0.0       0.0
## post2 -2.0       1.9
```

Income Does Not Have Much of an Effect (here)

```
plot(conditional_effects(post2, effects = "Income"))
```