STA365: Homework 2

Eric Zhu

02/04/2021

Contents

Quick Introduction

We are tasked with modelling student maths anxiety among university students. We are given access to "population" level data for student majors and corresponding genders. Anxiety is measured on a scale of 10-50, and the group was given a treatment ($Z = 1$) or a placebo ($Z = 0$). We're therefore trying to estimate the average treatment effect (ATE), given by:

$$\mathbb{E}(y \mid Z = 1) - \mathbb{E}(y \mid Z = 0)$$

Note that y is the difference between pre/post intervention scores.

Additionally, we consider a few variables for our model(s) in this homework: `gender`, `major`, `pre-intervention score`, and `treatment` indicator.

The model we want to consider is the MRP model, so we'll need to define which covariates are considered random and fixed.

Since our goal is to model the effect of intervention on university students rather than examining the effects of `gender` or `major` on maths anxiety. So then, it is sensible for `gender` and `major` to enter both models as random effects as we aim to "control" for these two variables.

For all model parameters, we will consider normal distributions for priors and if applicable truncated normal distributions. Additionally, we posit that the data is also normally distributed as qualities about animal populations such as weight, height, and in our case anxiety are often times normally distributed. We also justify this by the CLT given that populations are large and so a normal distribution seems sensible. So then if our experimental group are IID draws from the population, we should expect a normal distribution for the sample too. However, we do know that there are hard bounds on the maths anxiety score, so we consider a truncated normal likelihood for our response variable (maths anxiety score) on the bounds of (10, 50).

The first thing we'll note here is that we know very, very little about maths anxiety. In fact, the only information we have to work with is that maths anxiety scores are exactly on the interval [10,50], leading us to need a truncated normal distribution. So a sensible strategy for setting priors is to go with extremely weakly informative priors, which should give us a solid but not "overbearing" regularizing effect on our posterior distribution. We will deliberately target our priors such that we would expect to see anxiety scores on a range of [0, 60] if had normally distributed data instead of a truncated normal distribution. This should give us heavier tails for a truncated normal distribution, and reflects the intuition of a weakly informative prior: more spread out prior mass to regularize the posterior but to not over regularize the posterior.

Also note that for all priors, we will set τ by 2τ since our prior distributions are all normal or half-normal, and so $\pm 2\tau$ covers $\approx 95\%$ of the normal distribution density. In other words, we see very little density above or below $\pm 2\tau$ respectively. So we can consider the 95^{th} percentile ($+2\tau$) to be a sort of upper "bound" as we have in prior work, i.e., hw1 and a1. We will refer to this 95^{th} percentile intuition consistently as an upper "bound" for setting priors. So in placing priors we want to have the sum of $\mu + 2\tau$ over all priors to be about 60. A caveat for random effects is that we want our prior on the standard deviation to capture the highest possible standard deviation of the corresponding random effect.

In short, we want to set our priors to have large variances since that allows for a truncated normal distribution with heavier tails, which allows for a weaker regularizing effect. We do so by supposing that our distribution of data is not a truncated normal distribution, but rather a normal distribution that has a 5^{th} and 95^{th} percentile at 0 and 60 respectively. We do this specifically to allow for easier setting of priors by using the 95% rule and because thinking this way enables us to have wide priors that assign more density to the tails of the truncated normal distribution.

Finally, in all plots comparing the posterior distribution to the prior distribution for parameters, the darker blue histogram is the posterior distribution while the lighter blue is the prior distribution.

Modelling the pre-intervention scores

Our model for pre-intervention scores considers only the covariates `gender` and `major` with the response `score`. So using `lmer4` style syntax, we should consider the model:

$$\texttt{score} \sim \beta_0 + \texttt{(1|gender) + (1|major)}$$

So then our next goal is to put some priors on model parameters in order to obtain a posterior predictive distribution for the pre-intervention scores. In our model, we have an intercept (β_0), the coefficient for `gender` (β_{gender}), the random effect `major`. Note that for the random effect `major`, we have that each major gets an effect drawn from the distribution $N(0, \sigma^2)$, where σ^2 is the variance of the distribution for the random effect. Here, σ^2 is therefore a model parameters, which we need a prior on. Finally, we also have σ, the standard deviation of the data. To recap, we need to set normal (or half normal in the case of a standard deviation) priors for:

1. A prior distribution for the intercept, i.e., $\mu_{\beta_0}, \tau_{\beta_0}$.
2. A prior distribution for σ, i.e., μ_σ, τ_σ.
3. A prior distribution for the variance of our random effect `major`, i.e., τ_{major}.
4. A prior distribution for the variance of our random effect `gender`, i.e., τ_{gender}.

So then, we'll begin by considering the intercept (β_0). Since the range of maths anxiety scores range from at least 10 to at most 50, a sensible μ_{β_0} is 30, i.e., our prior belief for the intercept (grand mean) is that it is centred around 30. It seems sensible that $2 \cdot \tau_{\beta_0}$ be 5 because the grand mean being 35 or 25 is still very much in the middle of [10, 50]. So then we have the prior for β_0 to be $N(30, 2.5)$.

Next, we'll put a prior on the standard deviation for the two random effects. We do not have access to any information about how these effects interact with maths anxiety score, i.e., we do not have any information about effects size. In particular, we do not have information about which effect is bigger than the other, and so in the absence of such information to justify one effect size over another, we will give them the same really weakly informative prior on their standard deviations. This will allow the data to influence the posterior more. Since the prior on β_0 specifies that it is unlikely for $\beta_0 \geq 35$, we have that we still have about 25 maths anxiety points to before we hit 60, our "loose" targeted upper "bound". So if we suppose, for setting priors, that the at the 95^{th} percentile ($+2\sigma$, where σ is the standard deviation of a random effect) random effects each have an effect size of 10, then we find that if we had the variance of the random effects as fixed, we'd get a standard deviation of 5. Since we want a normally distributed prior on the standard deviation of the random effect, we can set the mean of the prior distribution to be 5. The standard deviation of the prior is a bit tricky since it can greatly affect the range of the effect size for our random effects. But a good value is 1 because then 2 times the standard deviation is just 2. So it would be unlikely to see standard deviations for the random effect greater than 7, meaning that it's unlikely to see a random effect size greater than 14. So incorporating both of our random effects, it'd be unlikely to see a random effect size greater than 28. While that's considerably larger than our original goal of 20, it's still alright since we truly are going for weakly informative priors that have large (but still sensible) spreads.

Finally, we need a prior on σ, the standard deviation of our response variable (pre-treatment anxiety score). Recall that we're trying to target approximately 60 as an upper "bound" on the anxiety score. Our upper "bound" for β_0 as described in the paragraph above is 35, and the combination of the random effects came out to approximately 28. So then we've got 63, which is over 60. So we'll take a bit of a different approach here. We'll first centre our prior for σ at 0, which captures the intuition that our model fits the data and also because there's no justification for thinking that people would arbitrarily deviate from the mean of the response variable. Next, we set our τ_σ to be 6 because $2\tau_\sigma$ would then be 12. So then our standard deviation for `score` could have an upper "bound" of 12. This seems sensible since the scale is from 10 to 50, so 12 is only a fraction of that. In other words, our belief that as an "upper bound" the natural variation in maths anxiety score on average 12 doesn't seem un-sensibly large given our limited information about maths anxiety scores.

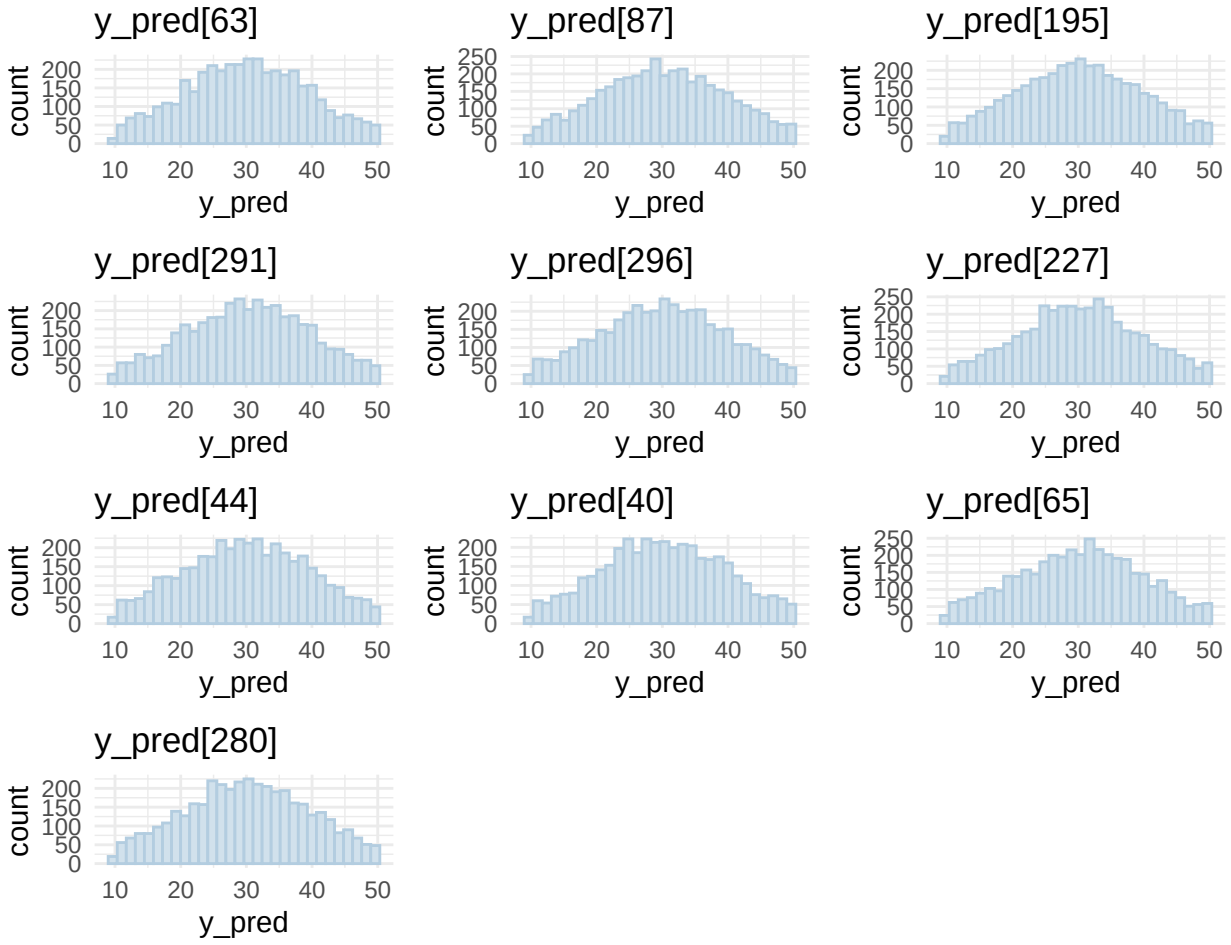
So to recap, we have these as our priors:

1. $\beta_0 \sim N(30, 5)$
2. $\sigma \sim N_+(0, 6)$
3. $\tau_{major} \sim N_+(5, 1)$
4. $\tau_{gender} \sim N_+(5, 1)$

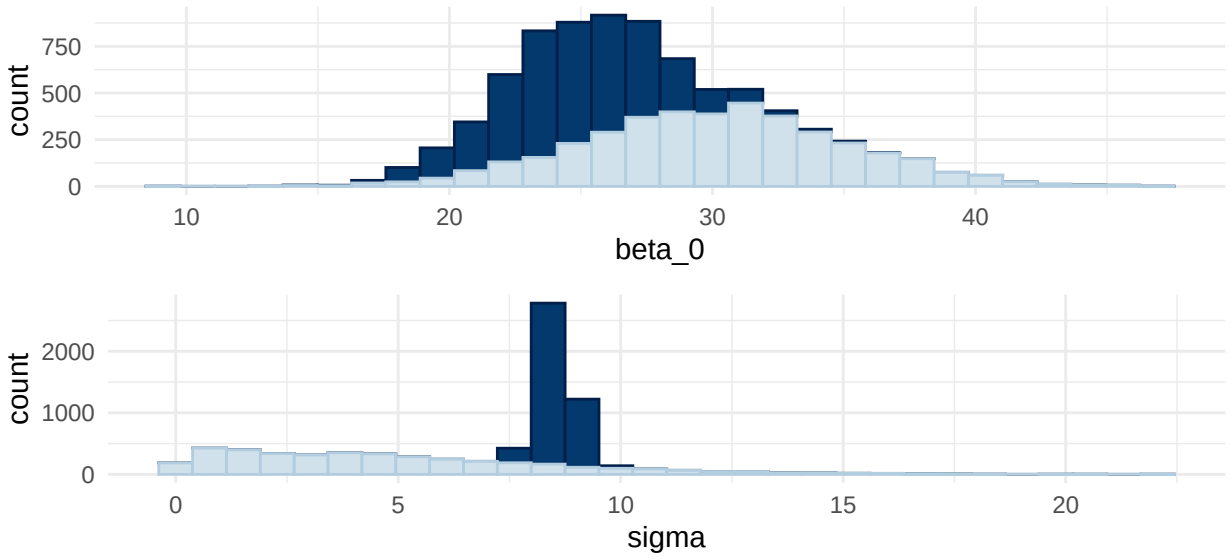
Additionally, we'll call the linear combination of our covariates to be μ. Each person in our data is given an entry of μ, e.g., the first person we observe will be `mu[1]`. μ is also the centre of our distribution that describes our likelihood, i.e., the truncated normal distribution.

Prior predictive checks/Comparison of prior and posterior distributions

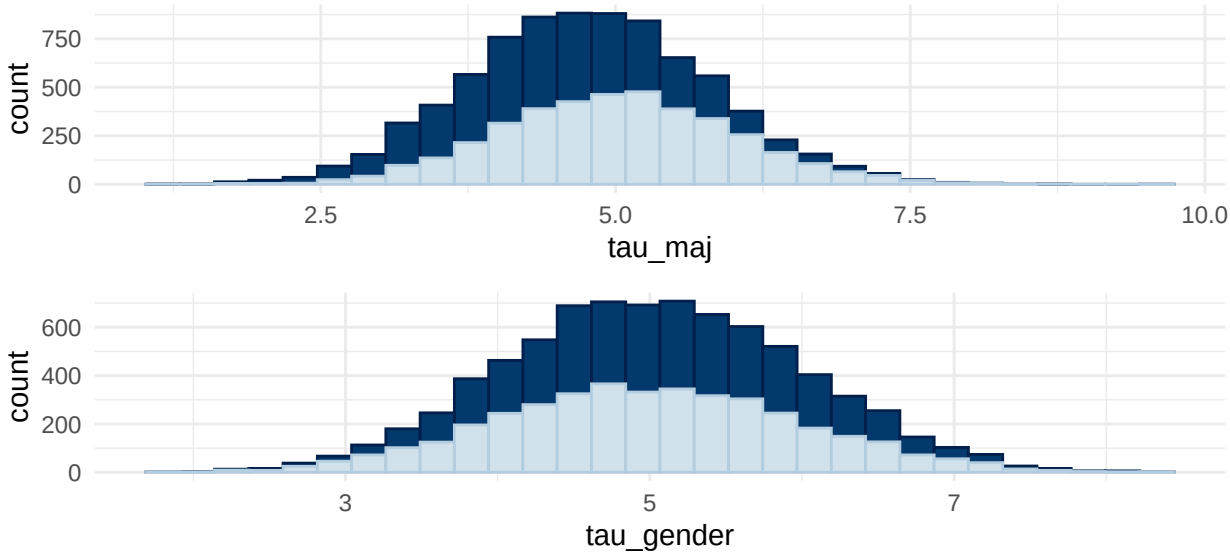
First we'll first check our prior predictive distribution:



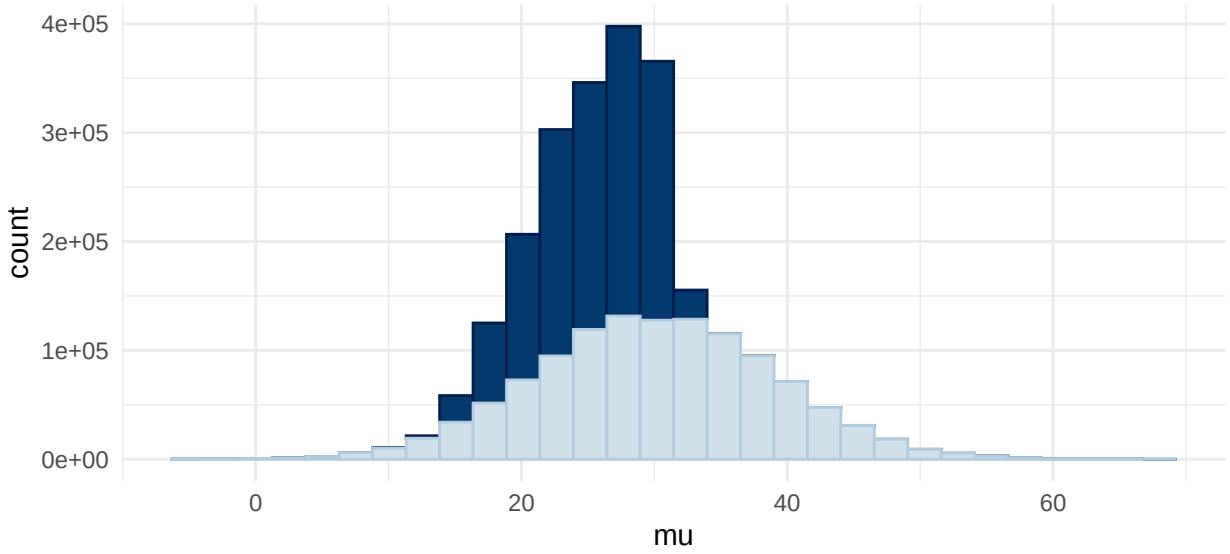
As we see above from 10 randomly sampled prior predictive distributions (corresponding to 10 different people), the distributions have generally pretty heavy tails as we'd expect from setting our priors. Our prior distributions are promising then to help regularize our posterior distributions but not overly so as we do not have much information about the maths anxiety scores. Our next step is to check the comparisons between the prior and posterior distributions:



From the two plots above, we see that in both cases, the posterior (darker blue) distribution contracts within the prior distribution (lighter blue). As such, we see the behaviour of a weakly informative prior. We also see that the prior distributions for both parameters have really long tails, which is exactly what we went for when setting priors (and conversely, the posterior distributions are far less spread out). And in particular, the prior distribution for σ is half normal distributed as we wanted. The comparison plot for β_0 is comparatively better than that of σ in terms of behaviour we'd expect from a weakly informative prior (and a well fit model) in that it contracts more with in the centre of the prior distribution. In fact, the centre of the posterior distribution for σ is not that close to 0 (at around 8), which is indicative that this model isn't a perfect fit for our data, but it's not too worrying as we don't have evidence for prior data conflict, i.e., a posterior distribution that is located where there is minimal (or no) prior distribution mass, and no model is perfect.

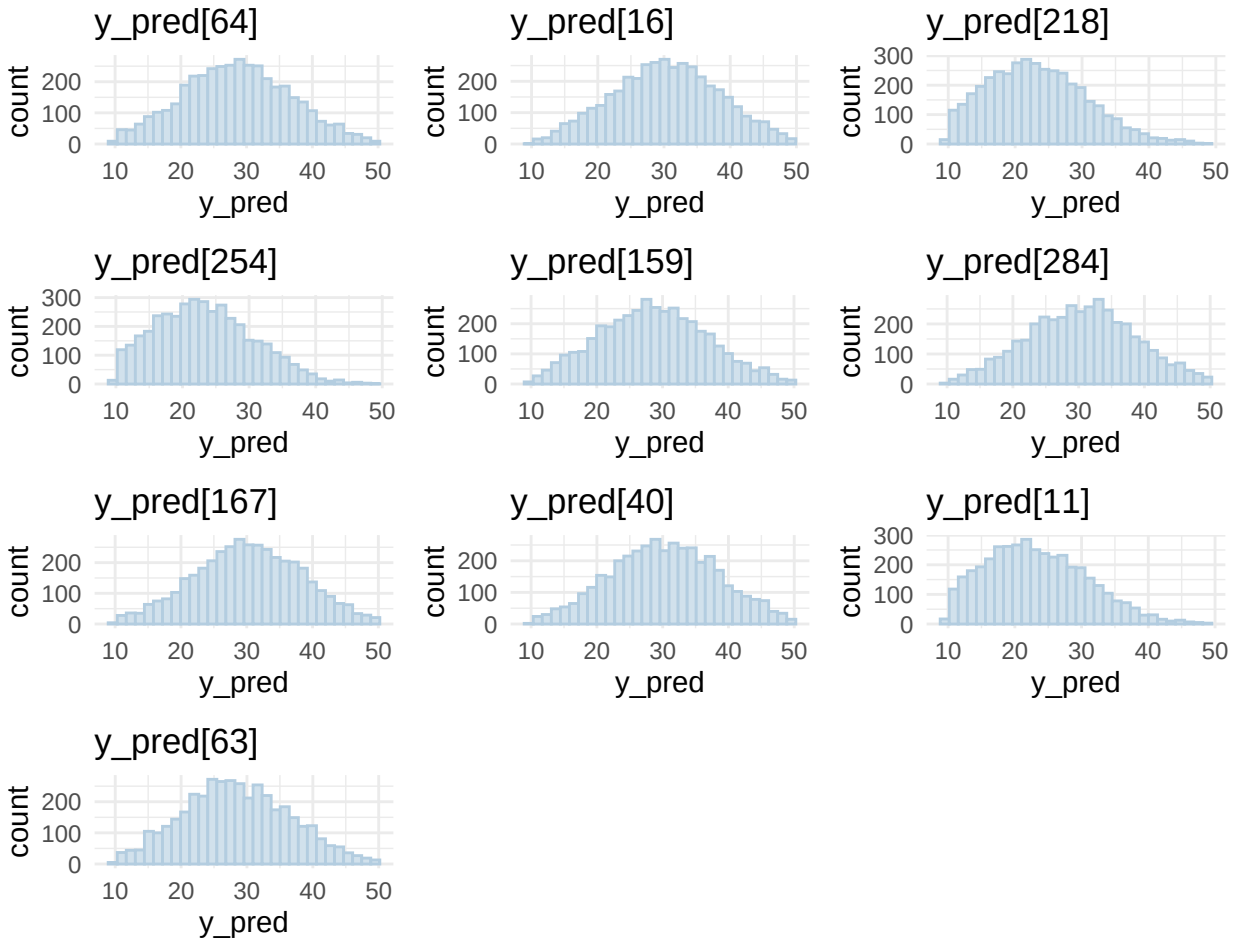


Next, both standard deviation parameters for our random effects look really good. The prior distribution again has long tails, which is what we purposely specified for really weakly informative priors, and the posterior distribution contracts well within the prior distribution. Additionally, the centre of the posterior distributions for both parameters are located very close to the centre of the prior distribution, especially with τ_{gender}. This not only gives us evidence that our model is a sensible model and one that fits decently but also that our priors are justified (no evidence whatsoever of prior-data conflict).

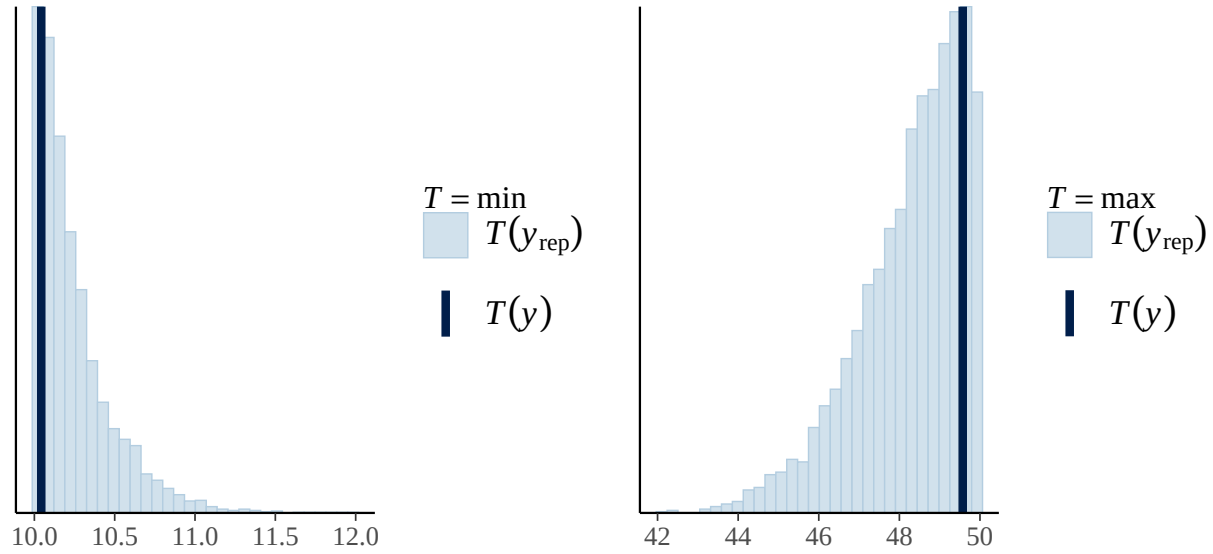


Finally, from examining the comparison between the prior and posterior distributions for μ, we find that it is more of what we have been observing with all of our other parameters. We find again: a prior distribution with long tails and a posterior distribution that contracts well within the prior distribution with a centre that is close to the centre of the prior distribution. These are all good signs of weakly informative priors and a model that fits the data decently well. We therefore conclude that we have justified priors for this model and that we have evidence that the model fit decently well (but not perfectly as we saw with the comparison plot of σ).

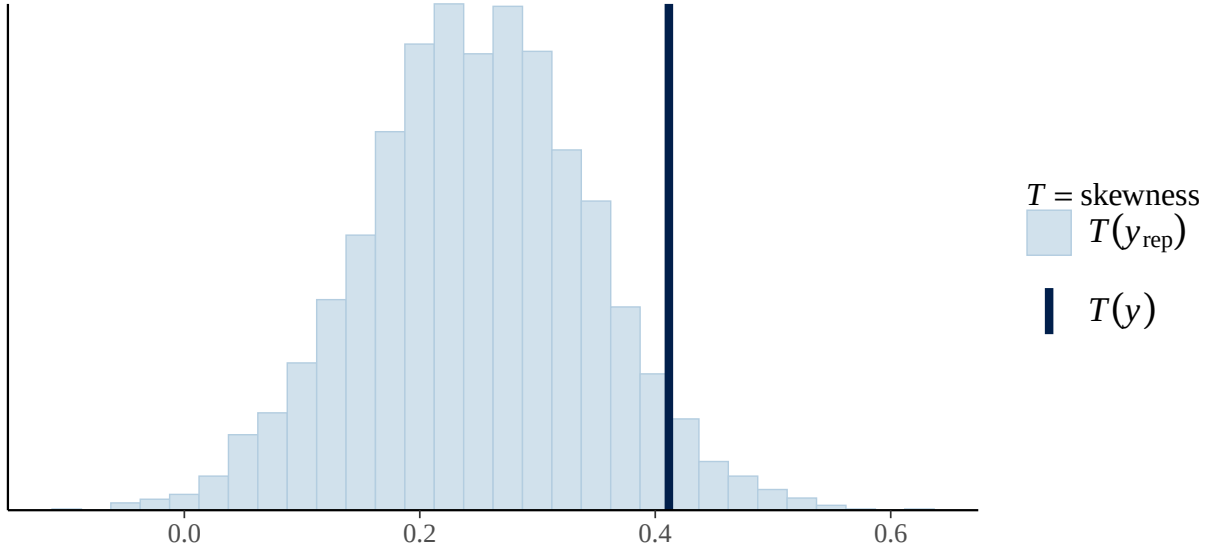
Posterior predictive distribution/Evaluation of test statistics



Our posterior predictive distribution looks how we'd expect it to look. We gave the prior distributions long tails so we'd also expect that the posterior predictive distribution to also have somewhat long tails, which we see here. The tails of course have far less density than those we observed in the prior predictive distribution.

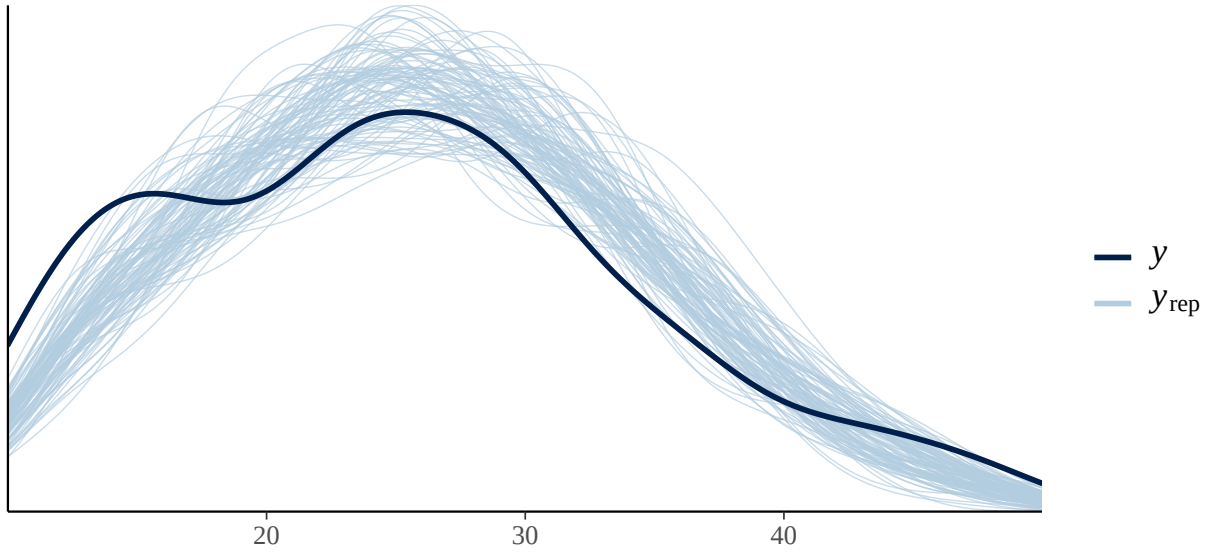


From examining our test statistic plots for `min`, `max` we see that the model was able to capture both test statistics. Both $T(y_{rep})$ distributions had long tails either on the right or left for `min` and `max` respectively. We'd actually expect to see this behaviour because we don't expect every posterior predictive distribution to have mass at the extremes of the anxiety score range, i.e., 10 and 50. For example, we wouldn't expect someone with extremely low maths anxiety (someone like a 4th year math specialist) to have a pre-intervention score of 50 in their posterior predictive distribution. So in fact, we do expect to see a sort of half normal distribution with somewhat long tails due to those individuals that may have a lot of maths anxiety or very little. But also note that both tails only range by about 2 for $T(y_{rep})$ of `min` and by about 6 for $T(y_{rep})$ of `max`. Both 1 and 6 are a fraction of the range of the score scale (40), and most importantly, the respective test statistic is located in the bulk of the mass for both $T(y_{rep})$ distributions. We conclude that this provides us evidence of a reasonable fit.

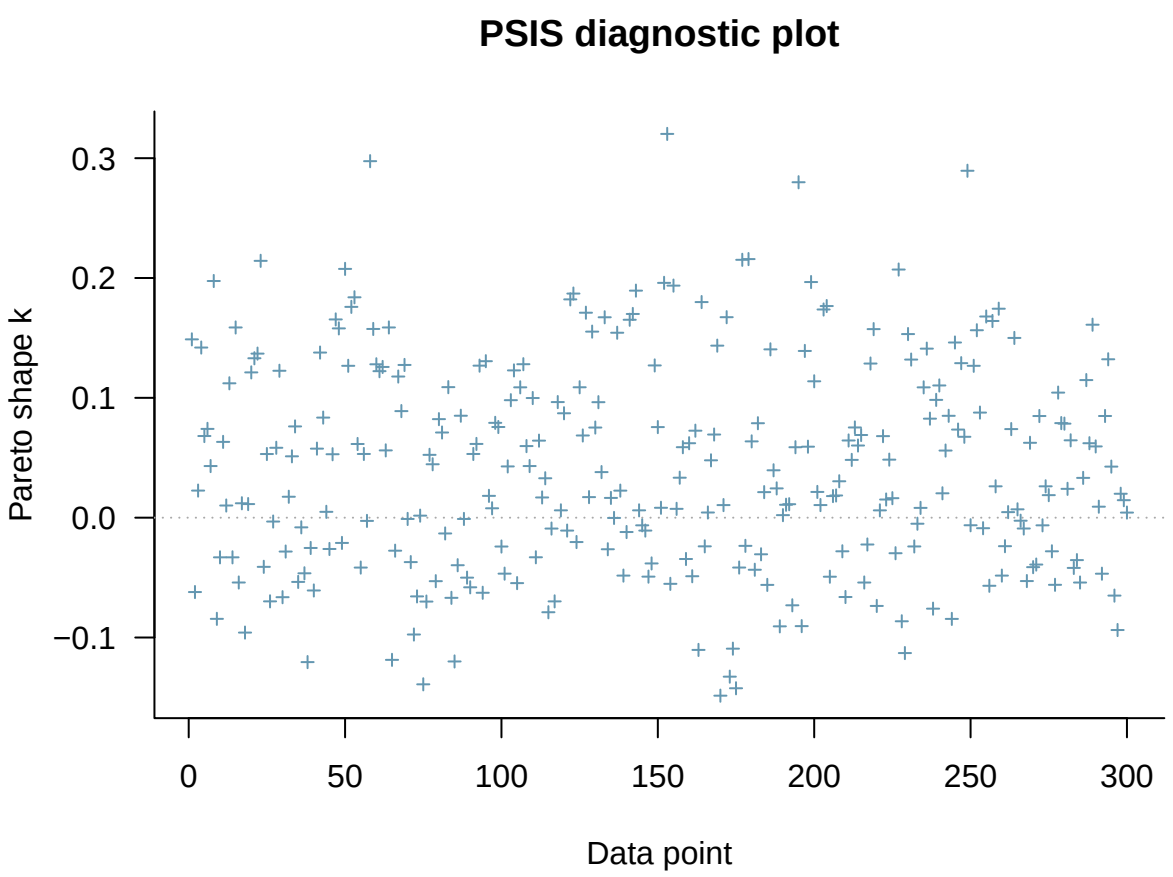


Finally, we see that the model was also fairly able to capture the `skewness` test statistic. The test statistic was captured more so on the right tail. But it wasn't egregiously on the end of the right tail, so we have evidence that the model fit the data well, but not extremely well, which we had evidence of from the posterior distribution of σ being centred somewhat far away from 0.

Evaluation of density plot/PSIS



From the plot above, we see that the model was poor at replicating the density of the y values around 10 to 20 by having consistently less density on this interval of y values (recall our possible y values range from 10 to 50). Additionally, the model was somewhat poor at replicating the density from 30 to 40 by having consistently more density of y values on this interval. Overall, the model was poor at replicating the behaviour of the density estimate of y, we can clearly see a bimodal curve from the y density estimate, while the y_{rep} density estimates were overall unimodal and centred around 25, which does match that of the second mode of y. There just is no "dip" in density of y values around 20, which is most problematic. So clearly, the model isn't a perfect fit, and we substantiate that claim with evidence from the `skewness` test statistic plot and the σ comparison plot.



Finally, from the PSIS plot, we see that all $\hat{k}$ values are well below 0.5, indicating that the model has decent predictive performance. Although there are a few points that are above 0.3, which is somewhat concerning. So clearly, the model isn't fantastic at predicting all of the dataset through loo-cv. This does back up our evaluation of the density estimation, `skewness` plot, and σ comparison plot. Overall though, the model was performed decently well with loo-cv, and we have no indication of any particularly influential points that would give us cause for concern over a misspecified or inappropriate model given that the bulk of our $\hat{k}$ are well within -0.1 to 0.2.

Conclusion

Our model for pre-intervention scores seems to be a sensible model with justified priors. We see that from the comparison of prior and posterior distribution for model parameters, from the test statistic plots (in particular `min` and `max` plots), and from the PSIS plot (most $\hat{k}$ values are between -0.1 and 0.2). However, we do also find that the model isn't an extremely good fit, given that the σ posterior distribution was not centred that close to 0, the `skewness` plot, and the density estimate plot which showed the short comings of how the model wasn't great at replicating the behaviour of the density estimate curve of y (in particular the modality).

Modelling the post-intervention scores

Our model for pre-intervention scores considers all of our covariates we defined in the "quick introduction". Note that `gender` and `major` are still random effects are we wish to control for them as we are not really interested in these covariates. Our response is still `score`, the anxiety score post treatment. So using `lmer4` style syntax, we should consider the model:

$$\texttt{score} \sim \beta_0 + \texttt{(1|gender) + (1|major) + pre-intervention score + treatment}$$

We will now put priors on the model parameters similarly as we did with pre-intervention scores. Unlike the pre-intervention scores model, we have two more parameters, one for the coefficient of `pre-intervention score` and one for the coefficient of `treatment`. In our model, we have an intercept (β_0), the coefficient for `gender` (β_{gender}), the random effect `major`. Note that for the random effect `major`, we have that each major gets an effect drawn from the distribution $N(0, \sigma)$, where σ is the standard deviation of the distribution for the random effect. Here, σ is therefore a model parameter, which we need a prior on like before. Finally, we also have a different σ, the standard deviation of the data. We will follow much of the same process as process for the pre-intervention model, and initially target the interval [0, 60] for the additive effects from our prior distributions to set really weakly informative priors (we still know barely any information about the distributions of our covariates and response).

To recap, we need to set normal (or half normal in the case of a standard deviation) priors for:

1. A prior distribution for the intercept, i.e., $\mu_{\beta_0}, \tau_{\beta_0}$.
2. A prior distribution for σ, i.e., μ_σ, τ_σ.
3. A prior distribution for the variance of our random effect `major`, i.e., τ_{major}.
4. A prior distribution for the variance of our random effect `gender`, i.e., τ_{gender}.
5. A prior distribution for the coefficient of `treatment`, i.e., $\mu_{\beta_Z}, \tau_{\beta_Z}$.
6. A prior distribution for the coefficient of `pre-intervention score`, i.e., $\mu_{\beta_{\texttt{pre-intervention score}}}, \tau_{\beta_{\texttt{pre-intervention score}}}$.

We have a solid starting point from the pre-intervention model, so we'll start with those, but note that we'll need to greatly decrease the standard deviations on the prior parameters from pre-intervention model else we run the risk of having extremely wide priors that provide too weak of a regularizing effect. Like pre-intervention model, we'll set the prior for σ, the standard deviation of the response, last.

So then, we'll begin by considering the intercept (β_0). Since the range of maths anxiety scores range from at least 10 to at most 50, a sensible μ_{β_0} is 30, i.e., our prior belief for the intercept (grand mean) is that it is centred around 30. It seems sensible that $2 \cdot \tau_{\beta_0}$ be 2.5 because the grand mean being 32.5 or 22.5 is still very much in the middle of [10, 50]. So then we have the prior for β_0 to be $N(30, 2.5)$.

Next, we'll put a prior on the standard deviation for the two random effects. We do not have access to any information about how these effects interact with maths anxiety score, i.e., we do not have any information about effects size. So with the same justification as the prior for the random effects of the pre-intervention model, we'll suppose that if at the 95^{th} percentile (i.e. $+2\sigma$, where σ is the standard deviation of a random effect) random effects each have an effect size of 5, then we find that if we had the standard deviation of the random effects as fixed, we'd get a standard deviation of 2.5. Note that this is half of effect size we assumed for the pre-treatment model, which we justify because we want to tighten the priors for the post intervention model as we don't want our priors' regularizing effects to be too weak. Also it seems sensible that after math help, effects like gender and major may affect maths anxiety less. So we set our prior mean to be 2.5 as we set normally distributed priors for these two parameters. Similar to the reasoning for the setting mean of these two parameters, we will halve the priors' standard deviation w.r.t their values in the pre-treatment model, i.e., $\frac{1}{2} = 0.5$. Thus, we expect to see standard deviations above 2 very rarely, and so we expect to see effect sizes greater than $2 \cdot 2 + 2.5 = 6.5$ very rarely. So incorporating both of our random effects, it'd be unlikely to see a random effect size greater than 13. So the effect size we have "left" for the two other fixed effects is $60 - 13 - 32.5 = 17.5$.

Considering that we have 17.5 units of anxiety score "left" before we hit our "target" of 60 and given we know nothing about the coefficients/correlations of `pre-intervention score`/`treatment`, we want to give

their respective terms the same effect size, i.e., $\frac{17.5}{2} = 8.75$. Note that the scales of these two covariates are extremely different, and we may in fact may not be able to use 8.75 as a strict "target" but more as guidance. Theoretically, `pre-intervention score` is continuous on the interval [10,50] as it is a maths anxiety score, while `treatment` is a two levelled indicator variable. We will however use normal distributions for both priors because of the CLT.

So then, we will set the prior for `pre-intervention score` first. Since we are setting the coefficient for this covariate, we have no information that the coefficient is either positive or negative, so we will set the mean of this prior to 0. Setting $\tau_{\beta\texttt{pre-intervention score}}$ is tricky because it is possible for `pre-intervention score` to be 50 or 10, so any prior variance could violate our 8.75 effect size "target" or [0, 60] "target" from earlier. It makes more sense to first consider the interpretation of coefficients in linear regression. If we have $\beta\texttt{pre-intervention score} = 1$ and $\texttt{pre-intervention score} = 50$ (with all else constant), we say that the response changes by $\beta\texttt{pre-intervention score} \cdot \texttt{pre-intervention score} = 1 \cdot 50 = 50$ on average. Even from this example the effect from the `pre-intervention score` covariate seems extremely drastic mostly because `pre-intervention score` is able to be at most 50. So then we could set the target from this term to be 8.75, which would require $2 \cdot \tau_{\beta\texttt{pre-intervention score}} = 0.175$. It follows that $\tau_{\beta\texttt{pre-intervention score}} = 0.0875$.

However, in our preliminary prior checks, we found that the posterior distribution for $\beta\texttt{pre-intervention score}$ contracted within the prior distribution but a bit too much on the tails of the prior distribution for our liking. So this behaviour indicated that the prior was too constrained (as we saw the beginnings of possible prior-data conflict), meaning that we would want to increase the spread of the prior distribution by a considerable amount. A good starting point was the bump up $\tau_{\beta\texttt{pre-intervention score}}$ by a factor of 4. In doing so, we saw a posterior distribution that was more satisfactory and is what we settled on for this homework assignment, i.e., the final value for $\tau_{\beta\texttt{pre-intervention score}}$ is $\tau_{\beta\texttt{pre-intervention score}} = 0.35$. We do however give up some regularizing power on this prior since we've now increased the standard deviation of the prior distribution by a factor of 4, and so we should just expect potentially wider distributions for especially for the posterior of $\beta_{\texttt{pre-intervention score}}$ and the predictive distributions (both prior and posterior).

Now we'll set a prior for `treatment` (`Z`). Since `Z` only takes the values 0, 1, we'll consider only the case when `Z`=1 because the effect of the additive `treatment` term is necessarily 0 if `Z`=0. Similarly to `pre-intervention score`, we know nothing about how `treatment` is correlated with the response, so we set the mean of this prior distribution to 0. Since we wish to target 8.75 as an upper "bound", we will have $8.75 = Z \cdot \beta_Z = Z \cdot (0 + 2 \cdot \tau_{\beta_Z}) = 4.375$, where $Z = 1$ and β_Z is the $\approx 95^{th}$ percentile value for β_Z.

Finally, we need a prior on σ, the standard deviation of our response variable (post-treatment anxiety score). Recall that we're trying to target approximately 60 as an upper "bound" on the anxiety score. Similarly to the pre-intervention model, our we have no "room" for σ because our effects add up to about 60 if we consider $\approx 95^{th}$ percentile values for all the prior distributions. Although, unlike the pre-intervention model it would be unlikely to see values above 60 if we had a normal distribution for `score`. We'll first centre our prior for σ at 0, which captures the intuition that our model fits the data and also because there's no justification for thinking that people would arbitrarily deviate from the mean of the response variable. So as we did with the pre-intervention model, we set our τ_σ to be 6 because $2\tau_\sigma$ would then be 12. So then our standard deviation for `score` could have an upper "bound" of 12. Again since the post-intervention score is on a range of 10 to 50, 12 is only a fraction of that. In other words, our belief that as an "upper bound" the natural variation in maths anxiety score on average 12 doesn't seem unsensibly large given our limited information about maths anxiety scores.

So to recap, we have these as our priors:

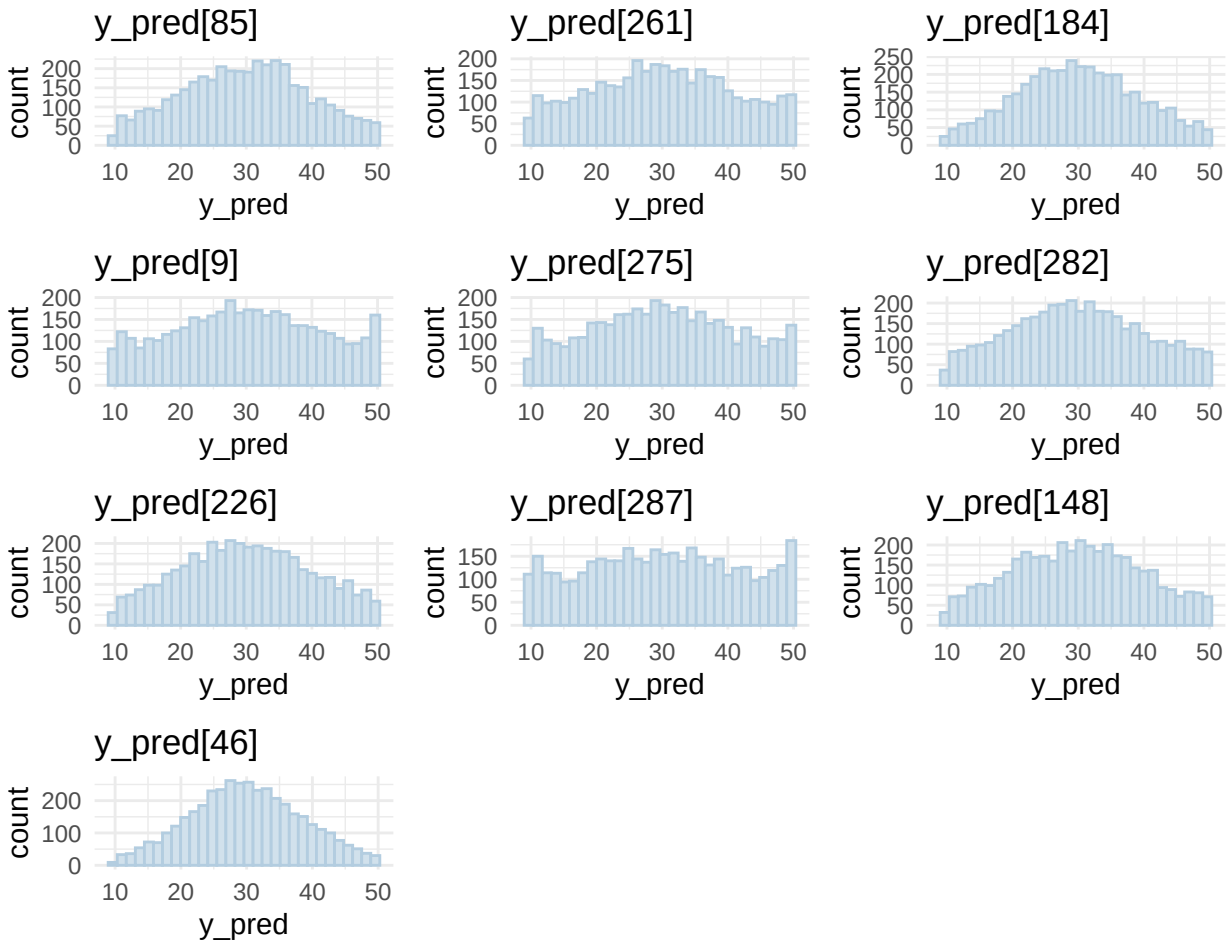
1. $\beta_0 \sim N(30, 2.5)$
2. $\sigma \sim N_+(0, 6)$
3. $\tau_{major} \sim N_+(2.5, 0.5)$
4. $\tau_{gender} \sim N_+(2.5, 0.5)$
5. $\beta_Z \sim N(0, 4.375)$
6. $\beta_{\texttt{pre-intervention score}} \sim N(0, 0.35)$

Additionally, we'll call the linear combination of our covariates to be μ. Each person in our data is given

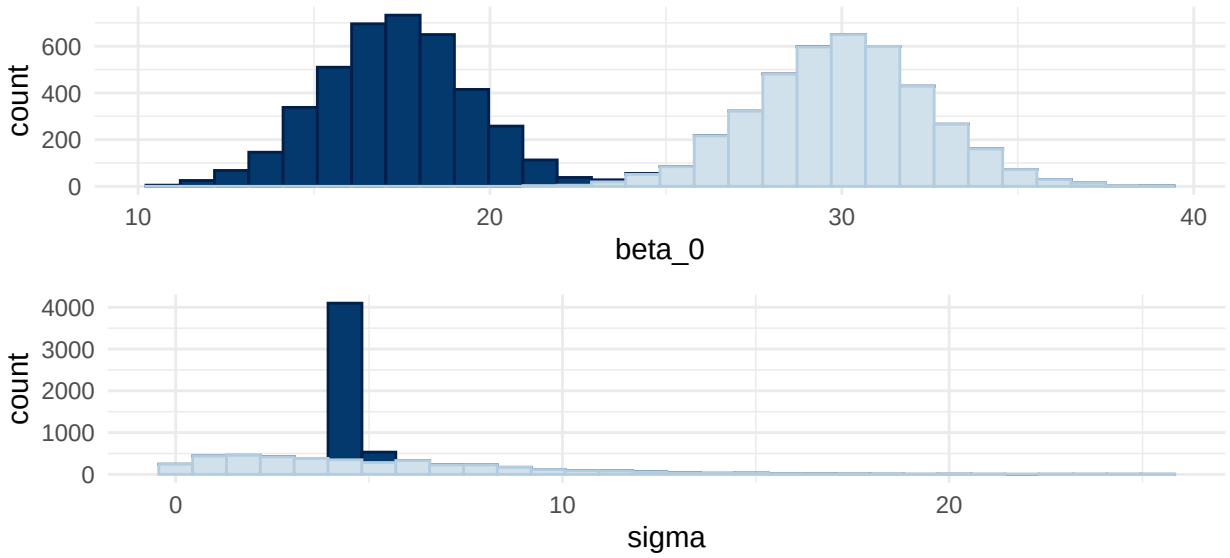
an entry of μ, e.g., the first person we observe will be `mu[1]`. μ is also the centre of our distribution that describes our likelihood, i.e., the truncated normal distribution.

Prior predictive checks/Comparison of prior and posterior distributions

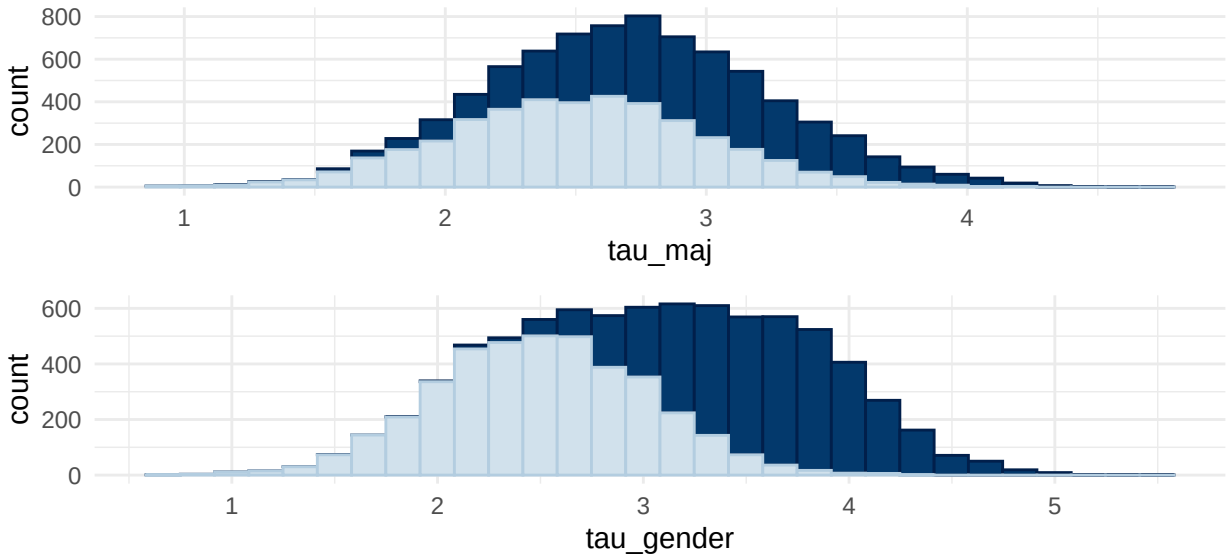
First we'll first check our prior predictive distribution:



As we see above from 10 randomly sampled prior predictive distributions (corresponding to 10 different people), the distributions still have fairly heavy tails as we'd expect from our "wide" priors, but in comparison to the pre-intervention model there is decidedly less mass on the tails of these distributions in general. There are a few "weird" distributions, i.e., with some distributions the tails have similar densities as the centre does. This is potentially explained by our really wide priors. Our prior distributions are promising then to help regularize our posterior distributions, potentially more so than the priors we had for the pre-intervention model. Our next step is to check the comparisons between the prior and posterior distributions:

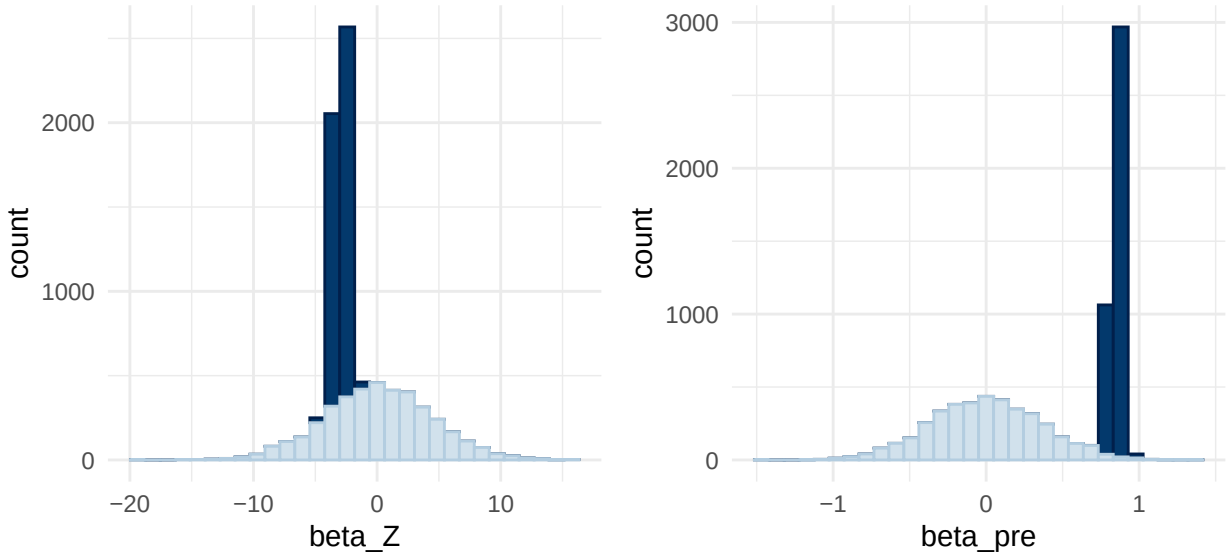


From the two plots above, we see that in both cases, the posterior (darker blue) distribution contracts within the prior distribution (lighter blue). As such, we see the behaviour of a weakly informative prior. We also see that the prior distributions for both parameters have really long tails, which is exactly what we went for when setting priors (and conversely, the posterior distributions are far less spread out). And in particular, the prior distribution for σ is half normal distributed as we wanted. The comparison plot for β_0 however does show that the model isn't a perfect fit as the posterior contracts more so on the left tail of the prior distribution. The posterior distribution is still rather spread out so, and the two distributions have a decent amount of overlapping mass, so it's not indicative of really any worrying prior-data conflict or a horrible fit. The posterior distribution for σ is (again like in the pre-intervention model) not that close to 0 (at around 5), which is indicative that this model isn't a perfect fit for our data. But like the pre-intervention model, it's not too worrying as we don't have evidence for prior data conflict, i.e., a posterior distribution that is located where there is minimal (or no) prior distribution mass, and no model is perfect.

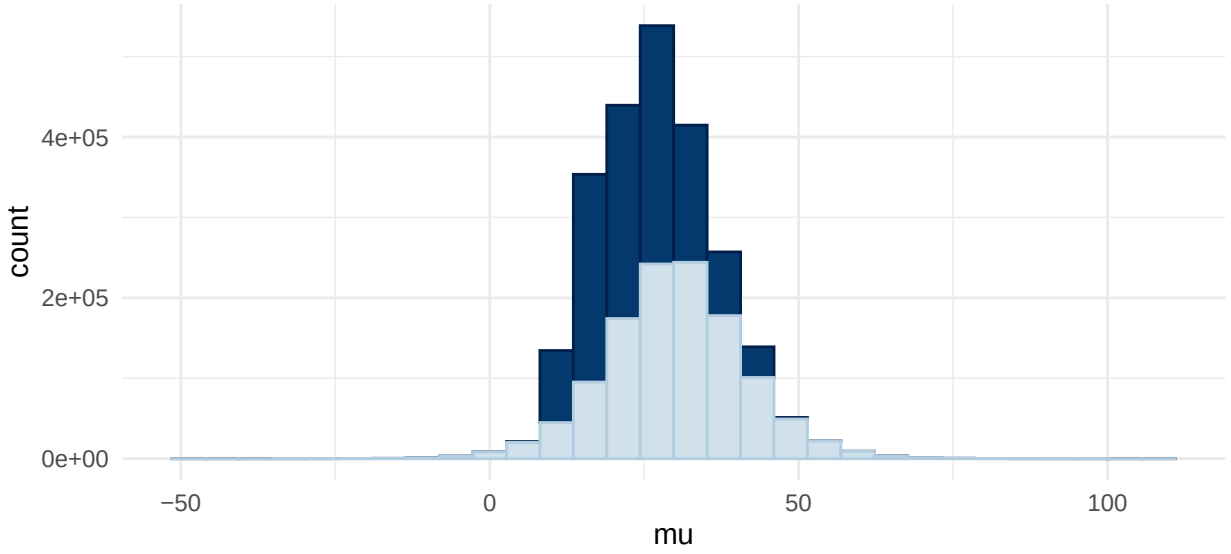


Next, both standard deviation parameters for our random effects look really good like the pre-intervention model. The prior distribution again has long tails, which is what we purposely specified for really weakly informative priors, and the posterior distribution contracts well within the prior distribution. Additionally, the centre of the posterior distributions for both parameters are located very close to the centre of the prior

distribution, especially with τ_{maj}. This not only gives us evidence that our model is a sensible model and one that fits decently but also that our priors are justified (no evidence whatsoever of prior-data conflict).



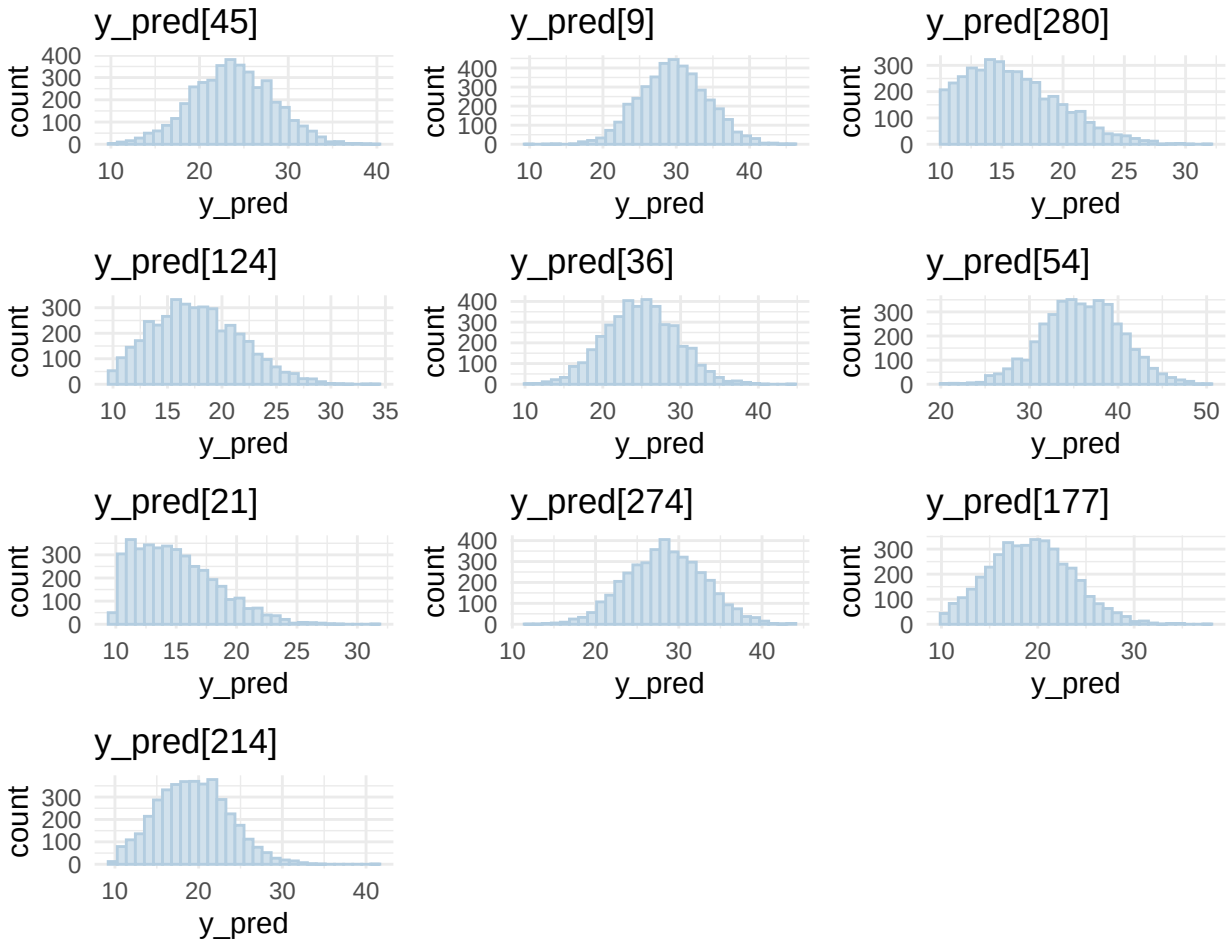
Next, the comparison plots of our fixed effects coefficients tell a sightly different story between them. The comparison plot for β_z, i.e., the coefficient for treatment, shows a well specified prior with the posterior contracting well within the prior and near the centre; these are all signs of a good fit too. Contrarily β_{pre} , the coefficient for the pre-treatment score is further away from the centre of the prior distribution. But it is worth noting that the prior distribution has extremely wide tails and the scale of the plot is not large: approximately from -2 to 2. So then, while the posterior distribution is sitting on the right tail, it doesn't seem like it was too misspecified as the posterior is located where there is still considerable prior mass. So we have evidence that the priors are justified but the model is potentially just not a fantastic fit for the data.



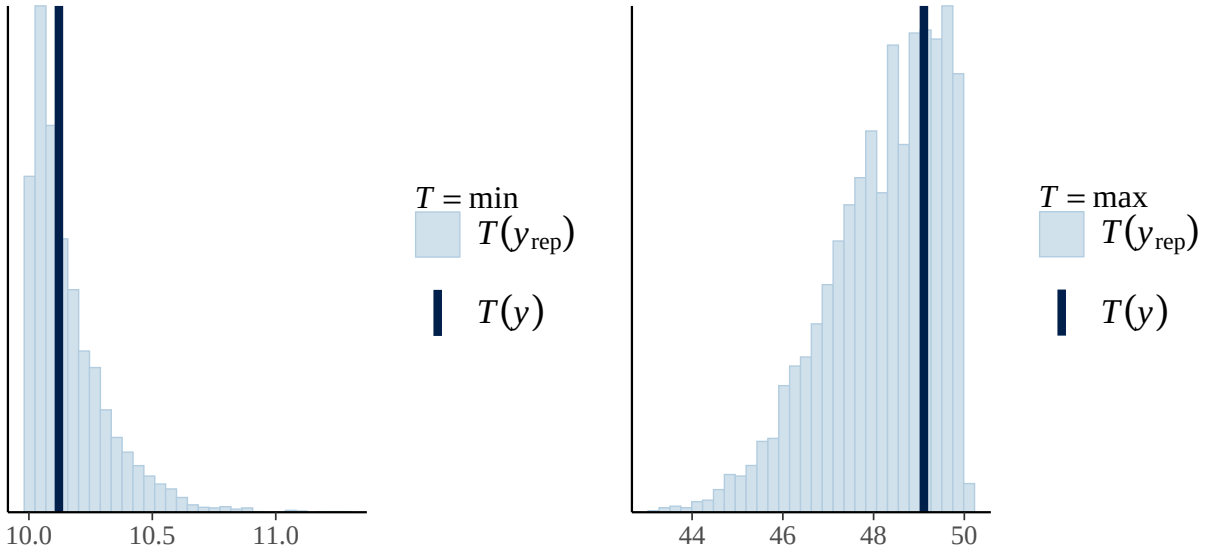
Finally, from examining the comparison between the prior and posterior distributions for μ, we find that it is more of what we have been observing with all of our other parameters. We find again: a prior distribution with long tails and a posterior distribution that contracts well within the prior distribution with a centre that is close to the centre of the prior distribution. These are all good signs of weakly informative priors and a model that fits the data decently well. We therefore conclude that we have justified priors for this model

and that we have evidence that the model fit decently well (but not perfectly as we saw with the comparison plot of σ).

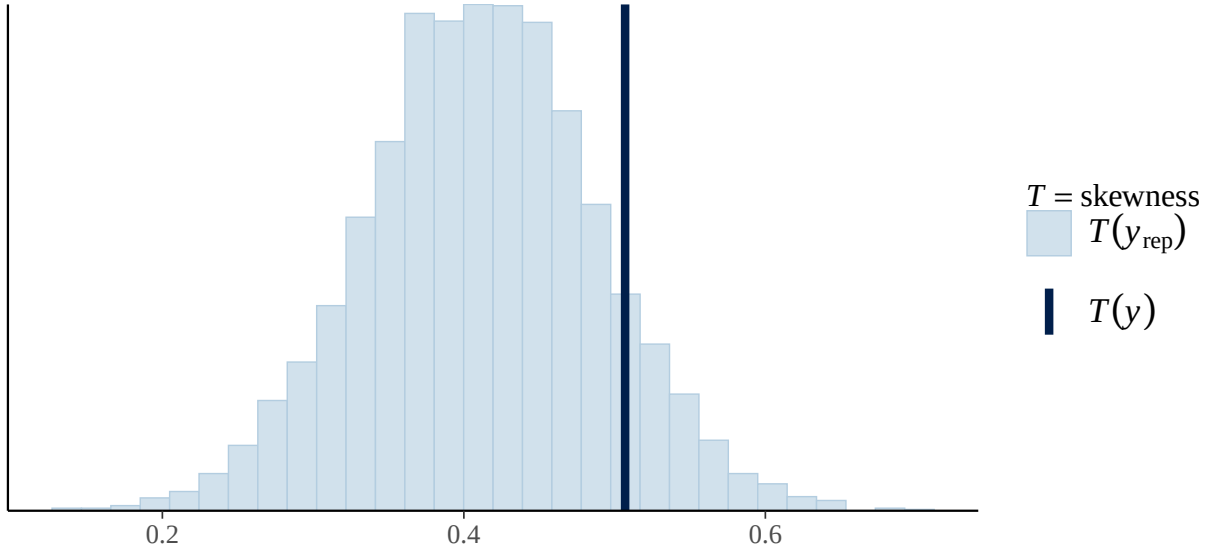
Posterior predictive distribution/Evaluation of test statistics



Our posterior predictive distribution looks how we'd expect it to look. We gave the prior distributions long tails so we'd also expect that the posterior predictive distribution to also have somewhat long tails, which we see here. The tails of course have far less density than those we observed in the prior predictive distribution. Also in comparison to the pre-intervention PPDs, we see that the distributions are generally more constrained, which may be reflective of our generally tighter priors in this model.

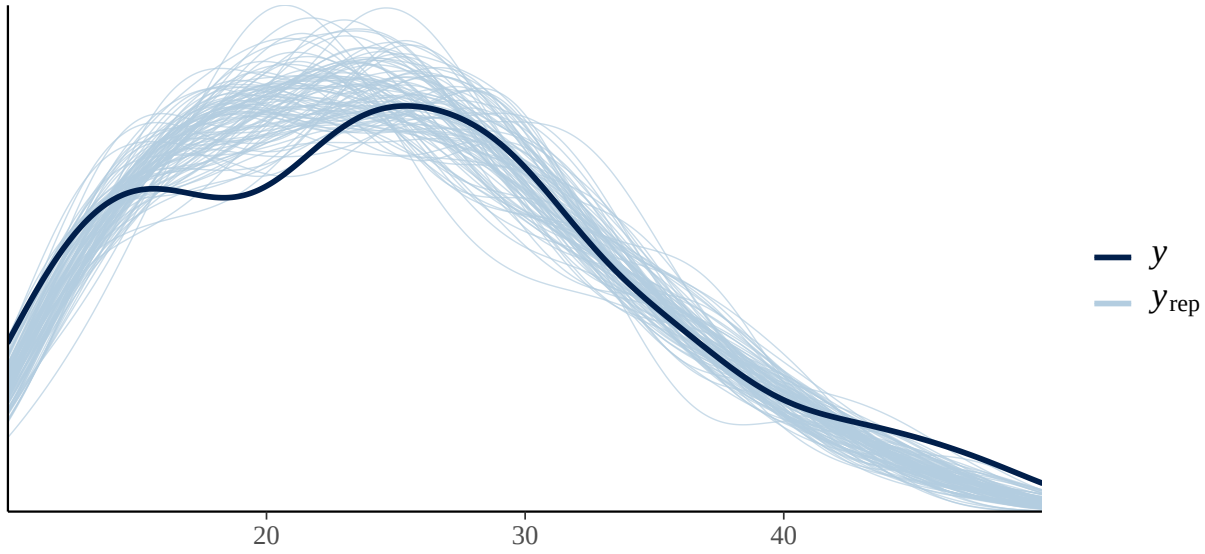


From examining our test statistic plots for `min`, `max` we see that the model was able to capture both test statistics. Both $T(y_{rep})$ distributions had long tails either on the right or left for `min` and `max` respectively. We'd actually expect to see this behaviour as discussed in the corresponding section for the pre-intervention model. Again, we wouldn't expect someone with extremely low maths anxiety (someone like a 4th year math specialist) to have a pre-intervention score of 50 in their posterior predictive distribution. So in fact, we do expect to see a sort of half normal distribution with somewhat long tails due to those individuals that may have a lot of maths anxiety or very little. But also note that both tails only range by about 2 for $T(y_{rep})$ of `min` and by about 6 for $T(y_{rep})$ of `max`. Both 1 and 6 are a fraction of the range of the score scale (40), and most importantly, the respective test statistic is located in the bulk of the mass for both $T(y_{rep})$ distributions. We conclude that this provides us evidence of a reasonable fit.

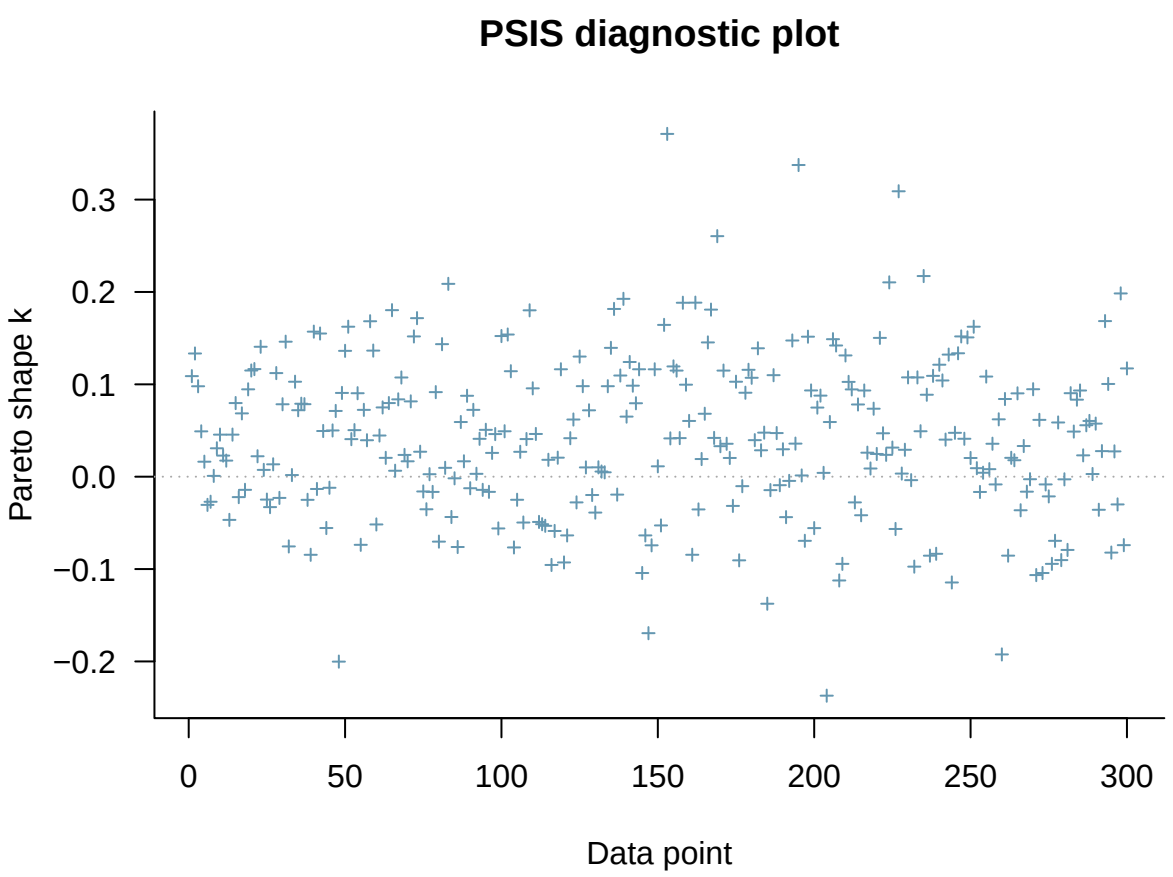


Finally, we see that the model was also fairly able to capture the `skewness` test statistic as it is fairly close to the centre of the $T(y_{rep})$ distribution at 0.4. This is indicative of the model not being a perfect fit, which we had evidence of from the posterior distribution of σ being centred somewhat far away from 0.

Evaluation of density plot/PSIS



From the plot above, we see that the model was poor at replicating the density of the y values around 15 to 25 by having consistently less density on this interval of y values (recall our possible y values range from 10 to 50). Overall, the model was poor at replicating the behaviour of the density estimate of y, we can clearly see a bimodal curve from the y density estimate, while the y_{rep} density estimates were overall unimodal and centred around 23, which matches none of the modes of the y density estimate curve. There just is no "dip" in density of y values around 20, which is most problematic. So clearly, the model isn't a perfect fit, and we substantiate that claim with evidence from the `skewness` test statistic plot, the σ comparison plot, and the β_{pre} comparison plot.



Finally, from the PSIS plot, we see that all $\hat{k}$ values are well below 0.5, indicating that the model has decent predictive performance. Although there are a few points that are above 0.3, which is somewhat concerning. So clearly, the model isn't amazing at predicting all of the dataset through loo-cv. This does back up our evaluation of the density estimation, $\beta_{\texttt{pre-intervention score}}$ comparison plot, and σ comparison plot. Overall though, the model was performed decently well with loo-cv, and we have no indication of any particularly influential points that would give us cause for concern over a misspecified or inappropriate model given that the bulk of our $\hat{k}$ are well within -0.2 to 0.2.

Conclusion

Our model for pre-intervention scores seems to be a sensible model with justified priors. We see that from the comparison of prior and posterior distribution for model parameters, from the test statistic plots, and from the PSIS plot (most $\hat{k}$ values are between -0.2 and 0.2). However, we do also find that the model isn't an extremely good fit, given that the σ posterior distribution was not centred that close to 0, the comparison plot for $\beta_{\texttt{pre-intervention score}}$, and the density estimate plot which showed the short comings of how the model wasn't great at replicating the behaviour of the density estimate curve of y (in particular the modality).

Estimating the ATE in the population

To estimate the ATE in the population we'll use multiple imputation to impute `pre-intervention scores` estimated for the population and ultimately construct a distribution of differences between pre and post intervention scores. Then we'll comment on the effectiveness on the intervention.

Estimating (predicting) `pre-intervention scores` for the population

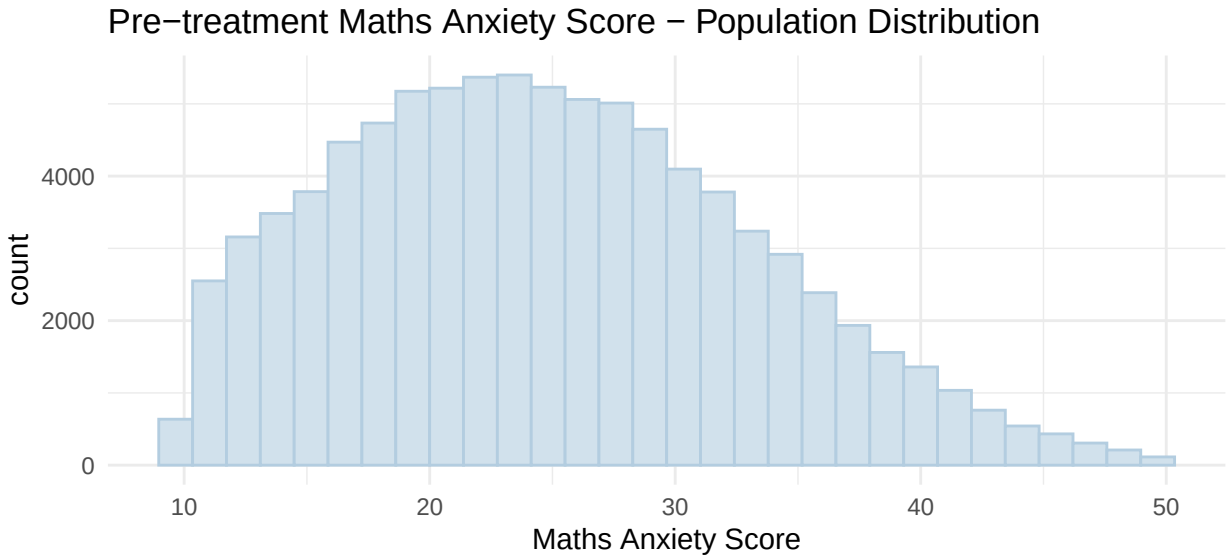
To predict the pre-intervention scores for the population we must marginalize over the model's parameter space so that we obtain a distribution only for `pre-intervention socres`. That is, we will consider every "class" (`major`/`gender` combination) in the population by looping over the population dataframe. We will then extract the rows from the `pre-intervention scores` PPD that correspond to that "class" and sampling 20 samples from that subset of the PPD. Should there not be a certain "class" in our experimental data (and thus not in the PPD), we will just not sample for that "class" because our model here isn't tasked with imputing missing "classes" nor is MRP necessarily designed for this job when we use it to predict the `pre-intervention scores`.

Here is the code for predicting the population `pre-intervention scores`:

```
# draw 20 samples for each population data point (major/gender combo)
# rows of the returned matrix are in order of those in the population df
# note that the returned matrix only includes combinations of maj/gender that exist in
# the experimental data
estimate_pre_popn <- function(joined_df, popn_df, ppd_pre, n_samples = 20){
  #browser()
  #print(paste("nrows: ", nrow(popn_df)))

  sample_matrix <- matrix(nrow = nrow(popn_df), ncol= n_samples)
  majors <- matrix(nrow = nrow(popn_df), ncol= 1)
  genders <- matrix(nrow = nrow(popn_df), ncol= 1)
  pop_ids <- matrix(nrow = nrow(popn_df), ncol= 1)

  matrix_index <- 1

  # get 20 scores for each population data point
  for(row in 1:nrow(popn_df)){
    # get matching experimental row indices for each popn data point
    # from joined_df
    popn_sample <- popn_df[row, ]
    #print(paste("row: ", row, " popn_sample: ", popn_sample))
    exp_row_idxs <- joined_df %>%
      filter(gender == popn_sample$gender
             & major == popn_sample$major) %>%
      select(exp_idx) %>%
      pull(exp_idx) %>%
      unique()
    #print(paste("row ", row,
    #            " has matching indexes of length: " ,
    #           length(exp_row_idxs)))


    # subset the ppd_pre matrix with corresponding rows
    # iff we are able to
    if(length(exp_row_idxs) > 0){
      ppd_subset <- ppd_pre[exp_row_idxs, ]
```

```
      # sample from resulting subset
      pre_treatment_sample <- sample(ppd_subset, n_samples)
      #print(class(post_treatment_sample))
      sample_matrix[matrix_index, ] <- pre_treatment_sample
      # append the index of the first element of the exp_row_idxs
      # corresponds to the row in the experimental dataframe
      # with this specific major/gender combo
      majors[matrix_index, ] <- popn_sample %>% pull(major)
      genders[matrix_index, ] <- popn_sample %>% pull(gender)
      pop_ids[matrix_index, ] <- popn_sample %>% pull(ID)
      matrix_index <- matrix_index + 1
    }
    #print(post_treatment_sample)
  }

  return(list(na.omit(sample_matrix), na.omit(majors), na.omit(genders), na.omit(pop_ids)))
}
```

```
matched_rows <- match_rows(exp_dat, pop_dat)
ret_list <- estimate_pre_popn(matched_rows,
                              pop_dat,
                              posterior_pd_pre)
pre_treatment_population <- ret_list[[1]]
majors <- ret_list[[2]]
genders <- ret_list[[3]]
pop_ids <- ret_list[[4]]
```



As a sanity check we have plotted the above plot. Our estimated population distribution for the pre-treatment maths anxiety score seems to be right skewed, and centred around 25. It seems to be more or less consistent with behaviour we observed with our posterior predictive checks. This makes sense since this distribution is a post stratified (predicted for each individual in the population) distribution, with the individuals integrated out (for all of those who we had data for in our experimental data).

Constructing the ATE distribution using predicted `post-intervention scores`

Now that we've gotten our population estimates for `pre-intervention scores`, we can impute them in order to predict the `post-intervention scores`. We do so by looping over the `pre-intervention scores` that we estimate to be the population, and using its corresponding class to predict `post-intervention scores` for both when `Z=1` and when `Z=0`. Recall that since the post-intervention model was fit using data that included `anxiety_after`, we don't have to explicitly impute the scores and refit the model for each new vector of pre-intervention scores as the weights for the other covariates are dependent on the post-intervention scores. We will in fact only need to sample a set of values for each of our model's terms, which we recall as:

$$\texttt{score} \sim \beta_0 + \texttt{(1|gender) + (1|major) + pre-intervention score + treatment}$$

This set will serve as our weights for predicting the `post-intervention scores` using our imputed values for `pre-intervention scores`. We will also only calculate the linear combination using these sampled weights and the imputed `pre-intervention scores` rather than drawing from a different posterior predictive distribution because like linear regression, we are interested in the difference in conditional expectation:

$$\mathbb{E}(y \mid Z = 1) - \mathbb{E}(y \mid Z = 0)$$

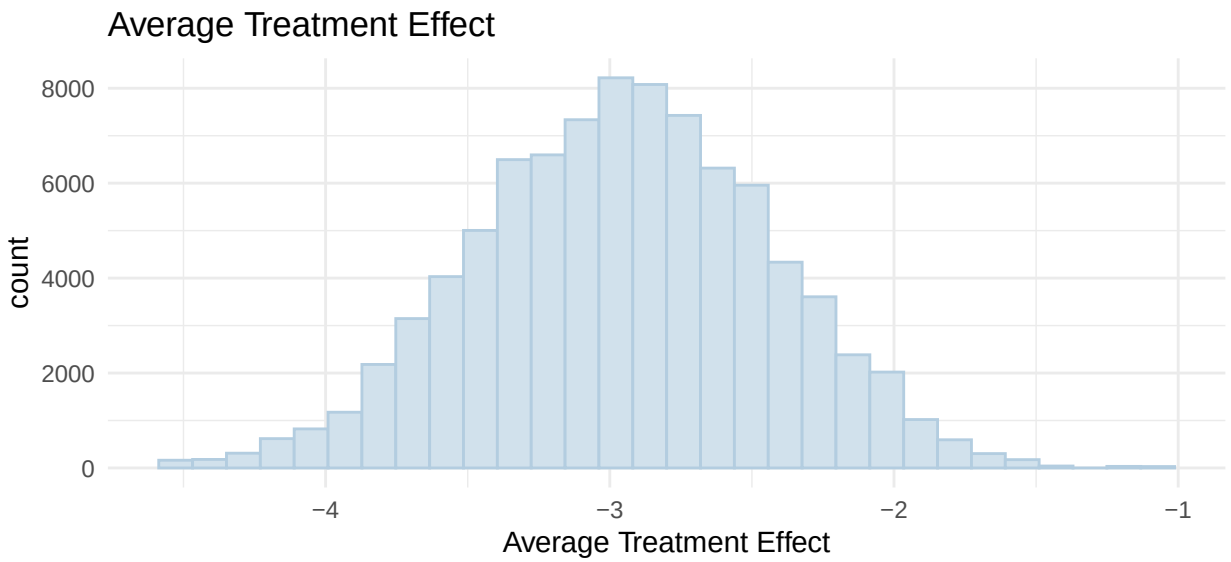
From linear regression knowledge, we know that our predictions from our model are considered as coditional expectations in the form of: $\mathbb{E}(y \mid x = a)$, where y, x, a are some real valued vectors. In effect, since we have more parameters, we'll want to marginalize over both the "classes" and the `pre-intervention scores` as those are parameters our predictions are conditional on. We do so by sampling a constant number of samples for each model parameter for every individual of the population (corresponding to that individual's "class"), and this process will marginalize over the class and pre-intervention scores. Additionally it'll perform our post-stratification given that we have a small, finite population. To obtain expectations conditioned on treatment status, we will first predict score when `Z=1` and then predict scores when `Z=0`. The sampled parameters will be the same as we wish to examine the additive effect of the treatment, and having enough samples (20+) for parameters will sufficiently marginalize over parameter space for our model's parameters. Then we'll be able to subtract all values from these two sets of predictions as required, leaving us with our desired ATE.

Here's the code for the multiple imputation process and ATE construction process:

```
# do multiple imputation for post-treatment using pre intervention scores
# draw 20 samples for each population data point (major/gender combo)
estimate_post_popn <- function(joined_df, majors, genders,pop_ids, pre_int_popn,
                               post_draws,
                               n_samples = 20){

  #browser()
  ate_samples <- matrix(nrow = nrow(pre_int_popn), ncol= n_samples)
  treatment_samples <-  matrix(nrow = nrow(pre_int_popn), ncol= n_samples)
  placebo_samples  <- matrix(nrow = nrow(pre_int_popn), ncol= n_samples)
  #print(paste("nrows: ", nrow(popn_df)))

  beta_0 <- subset_draws_df(post_draws, c("beta_0"))
  beta_pre <- subset_draws_df(post_draws, c("beta_pre"))
  beta_Z <- subset_draws_df(post_draws, c("beta_Z"))
  sigma <- subset_draws_df(post_draws, c("sigma"))
  u_major1 <- subset_draws_df(post_draws, c("u_major[1]"))
  u_major2 <- subset_draws_df(post_draws, c("u_major[2]"))
  u_major3 <- subset_draws_df(post_draws, c("u_major[3]"))
  u_major4 <- subset_draws_df(post_draws, c("u_major[4]"))
  u_major5 <- subset_draws_df(post_draws, c("u_major[5]"))
  u_gender1 <- subset_draws_df(post_draws, c("u_gender[1]"))
```

```
  u_gender2 <- subset_draws_df(post_draws, c("u_gender[2]"))
  u_gender3 <- subset_draws_df(post_draws, c("u_gender[3]"))

  random_effects_df <- data.frame(u_major1, u_major2,u_major3, u_major4,
                                  u_major5, u_gender1, u_gender2,
                                  u_gender3)

  # loop over the members of population we were able to predict for
  for(row in 1:nrow(pre_int_popn)){

    # find these parameters (major/gender)
    # should only be one combination
    # the set of all correct major, gender, Z combos
    # for each entry in column
    major <- majors[row,1]
    gender <- genders[row, 1]
    len <- n_samples


    # use the betas from the post-intervention model
    # and the predicted population pre-intervention scores
    # as covariates
    # recall that our model is:
    # y = beta_0 + beta_pre * pre_intervention + beta_Z * Z
    # + u_major + u_gender

    placebo <-  DescTools::Sample(beta_0, len) +
        DescTools::Sample(beta_pre, len) * pre_int_popn[row, ] +
        subset_helper(random_effects_df, "u_major", major, len) +
        subset_helper(random_effects_df, "u_gender", gender, len)

    # add the treatment effect to the "base" placebo prediction

    treatment <- placebo +
        DescTools::Sample(beta_Z, len) * 1

    placebo <- placebo %>% pull(beta_0)
    treatment <- treatment %>% pull(beta_0)


    ate_samples[row, ] <- treatment-placebo
    treatment_samples[row, ] <- treatment
    placebo_samples[row, ] <- placebo

  }

  return(list(ate_samples, treatment_samples, placebo_samples))
}

# n_samples corresponds to the nrows of the population
subset_helper <- function(draws_df, name, value, n_samples=300){
  pasted <-  paste(name,".",value,".", sep="")
  subset <- draws_df %>% select(pasted)
```

```
  #subset <- subset_draws_df(draws_df,pasted)
  return(DescTools::Sample(subset, n_samples))
}
```



From the plot of the Average Treatment Effect, we see that there is a clear effect from the maths anxiety treatment, i.e., those who had the treatment and not the placebo saw an decrease in maths anxiety. In fact, since 0 (and no positive numbers) were not in the distribution, we can safely say that this treatment was a successful one! One observation of note is that the effect size isn't huge, i.e., only at most about -5, which is just a fraction of the 10-50 scale (at most this treatment reduced maths anxiety by about 10%). But perhaps comparatively, this effect is large; we just don't know due to a lack of data.

Although it is worth noting that since our model only applies to the population we were given, i.e., these ≈ 4000 university students, we would need to refit the model and use a different post stratification strategy given a potentially different population. The biggest drawback of this approach was that we were able to post stratify by just predicting for each individual in the population as our population was a small finite number.

Appendix

R code

Helpers

Posterior vs prior plots

```
plot_prior_posterior <- function(prior_fit, posterior_fit, variable_name=""){

  posterior_plot <-melt(as_draws_matrix(subset_draws(posterior_fit,
                                                     regex =TRUE,
                                                     variable = variable_name))) %>%
    mutate(variable = str_replace_all(variable,
                                      pattern=paste(variable_name,".*", sep=""),
                                      replacement = "posterior"))
  prior_plot <- melt(as_draws_matrix(subset_draws(prior_fit, regex =TRUE,
                                                  variable = variable_name))) %>%
    mutate(variable = str_replace_all(variable,
                                      pattern=paste(variable_name,".*", sep=""),
                                      replacement = "prior"))

  comparison_df <- rbind(prior_plot, posterior_plot)

  comparison_plot <- ggplot(comparison_df,aes(x=value,
                                              fill = variable,
                                              color = variable)) +
    geom_histogram(alpha=1) +
    scale_fill_manual(values=c(color_scheme_get()$dark,
                               color_scheme_get()$light)) +
    theme_minimal() +
    theme(legend.position="none") +
    labs(x=variable_name)+
    scale_color_manual(values =c(color_scheme_get()$dark_highlight,
                                 color_scheme_get()$light_highlight))

  return(comparison_plot)

}
```

Generate predictive distribution for truncated normal

```
# each row corresponds to a sample from the population ("individual")
# so each row corresponds to the row from the experimental data
# with the same index.
# each column corresponds to an MCMC sample

# this runs in O(n*m), where n is the number of iterations
# m is the number of "observations"/"samples" - this is slow
tn_pred_distn <- function(fit_draws, upper_bound = 50, lower_bound = 10){
  # extract the mu, sigma row wise
  draws_df <- as_draws_df(subset_draws(fit_draws))
  mu_df <- subset_draws_df(draws_df, c("mu"))
  sigma_df <- subset_draws_df(draws_df, c("sigma"))
  #browser()
  # n by m matrix of ppd draws, where n is the number of observations in the data
```

```
  # m is the number of iterations (draws from the posterior in Stan)
  # so ppd_matrix[1, ] should be the predictions corresponding to row 1 in design matrix
  ppd_matrix <- matrix(nrow = ncol(mu_df), ncol= nrow(mu_df))
  # iterate over n sigma values, 1 for each iteration
  for(i in 1:nrow(sigma_df)){
    # get the corresponding sigma value for the iteration
    sigma <- sigma_df[i, 1]
    for(j in 1:ncol(mu_df)){
        # get mu
        # ith row (iteration) and jth column (experiment member)
        mu <- mu_df[i, j]
        # get draw using rtruncnorm
        pred <- rtruncnorm(1, lower_bound, upper_bound, mu, sigma)
        ppd_matrix[j, i] = pred
    }
  }
  return(ppd_matrix)
}
```

```
# each row corresponds to a sample from the population ("individual")
# so each row corresponds to the row from the experimental data
# with the same index.
# each column corresponds to an MCMC sample

# this is the semi vectorized version of the above code
tn_pred_distn_vec <- function(fit_draws, upper_bound = 50, lower_bound = 10){
  # extract the mu, sigma row wise
  draws_df <- as_draws_df(subset_draws(fit_draws))
  mu_df <- subset_draws_df(draws_df, c("mu"))
  sigma_vec <- as.matrix(subset_draws_df(draws_df, c("sigma")))
  #browser()
  # n by m matrix of ppd draws, where n is the number of observations in the data
  # m is the number of iterations (draws from the posterior in Stan)
  # so ppd_matrix[1, ] should be the predictions corresponding to row 1 in design matrix
  ppd_matrix <- matrix(nrow = ncol(mu_df), ncol= nrow(mu_df))
  for(j in 1:ncol(mu_df)){
      # get mu
      # ith row (iteration) and jth column
      mu <- as.matrix(mu_df[, j])
      # get draw using rtruncnorm
      pred <- rtruncnorm(1, lower_bound, upper_bound, mu, sigma_vec)
      ppd_matrix[j, ] = pred
    }
  return(ppd_matrix)
}
```

Extract desired columns from Stan draws df

```
subset_draws_df <- function(draws_df, column_names = c()){
  return(draws_df%>%select(starts_with(column_names)))
}
```

Plot PPD matrix

```
ppd_plot <- function(ppd_matrix, n_plots=10, file_name = "posterior_pre",
                     width = 3, height = 2, units = "in"){
    indicies <- sample(1:nrow(ppd_matrix), n_plots, replace = F)
    plot_list <- vector("list", n_plots)
    plot_list <- lapply(indicies, plot_helper, data = ppd_matrix)
    return(grid.arrange(grobs = plot_list))
}
```

Plot helper function

```
plot_helper <- function(i,data){
  title <- paste("y_pred[", i,"]",sep="")
  #print(title)
  y_pred <- data[i,]
  plot_tmp <- ggplot()+ aes_string("y_pred") +
        geom_histogram(alpha=1,
                       fill=color_scheme_get()$light,
                       color=color_scheme_get()$light_highlight)+
        labs(title=title)+
        theme_minimal()
  #print(plot_tmp)
  return(plot_tmp)
}
```

Join the experimental and population DataFrames

```
# get matching row from experimental data for each row in popn data
match_rows <- function(exp_data, popn_data){
  # experimental data has messed up rows
  #exp_ridx<- index(exp_data)
  exp_data <- exp_data %>% mutate(exp_idx = index(exp_data))
  popn_data <- popn_data %>% mutate(popn_id = ID)
  joined_df <- inner_join(exp_data, popn_data, by = c("major", "gender")) %>%
    select(-c("ID.x", "ID.y"))
  return(joined_df)
}
```

Markdown R code

```
library(cmdstanr)
library(loo)
library(tidyverse)
library(posterior)
library(bayesplot)
library(latex2exp)
library(reshape2)
library(gridExtra)
library(PerformanceAnalytics)
library(knitr)
library(deSolve)
library(R.utils)
library(genpwr)
library(rlist)
library(truncnorm)
library(DescTools)
```

```
register_knitr_engine(override = TRUE)
set.seed(365)

pop_dat <- readRDS("population.rds")

exp_dat <- readRDS("experimental_data.rds")

pre_intervention_dat <- exp_dat %>% select(-c(anxiety_after,Z))
post_intervention_dat <- exp_dat

# pre intervention code

data_list_pre <- list(N=length(pre_intervention_dat$anxiety_before),
                      y=pre_intervention_dat$anxiety_before,
                      J_maj = pre_intervention_dat$major%>%n_distinct(),
                      J_gender=pre_intervention_dat$gender%>%n_distinct(),
                      gender = pre_intervention_dat$gender,
                      major = pre_intervention_dat$major,
                      upper = 50,
                      lower = 10,
                      mu_sigma=0,
                      tau_sigma=6,
                      mu_intercept = 30,
                      tau_intercept = 5,
                      mu_tau_gender = 5,
                      tau_tau_gender = 1,
                      mu_tau_maj = 5,
                      tau_tau_maj = 1,
                      only_prior=1)

pre_intervention_mod <- cmdstan_model("pretreatment.stan", compile = TRUE)

data_list_pre$only_prior = 1
pre_intervention_prior_fit <- pre_intervention_mod$sample(data_list_pre,
                                                          seed = 365,
                                                          refresh = 500,
                                                          parallel_chains = 4,
                                                          adapt_delta = 0.99)
prior_pre <- pre_intervention_prior_fit$draws()
prior_pd_pre <- tn_pred_distn(prior_pre)
prior_pd_plots <- ppd_plot(prior_pd_pre)

data_list_pre$only_prior = 0
pre_intervention_fit <- pre_intervention_mod$sample(data_list_pre,
                                                    seed = 365,
                                                    refresh = 0,
                                                    parallel_chains = 4)
posterior_pre<-pre_intervention_fit$draws()

intercept_comparison <- plot_prior_posterior(prior_pre,
                                             posterior_pre, "beta_0")
tau_maj_comparison <- plot_prior_posterior(prior_pre,
                                           posterior_pre, "tau_maj")
```

```
tau_gender_comparison <- plot_prior_posterior(prior_pre,
                                              posterior_pre, "tau_gender")
sigma_comparison <- plot_prior_posterior(prior_pre,
                                         posterior_pre, "sigma")
mu_comparison <- plot_prior_posterior(prior_pre,
                                      posterior_pre, "mu")

grid.arrange(intercept_comparison, sigma_comparison)

grid.arrange(tau_maj_comparison, tau_gender_comparison)

mu_comparison

posterior_pd_pre <- tn_pred_distn(posterior_pre)

posterior_pd_plots <- ppd_plot(posterior_pd_pre)

min_pre <- ppc_stat(y = data_list_pre$y, yrep= t(posterior_pd_pre), stat = "min")
max_pre <- ppc_stat(y = data_list_pre$y, yrep= t(posterior_pd_pre), stat = "max")
skewness_pre <- ppc_stat(y = data_list_pre$y, yrep= t(posterior_pd_pre), stat = "skewness")
grid.arrange(min_pre, max_pre, ncol = 2)

skewness_pre

sample_rows <- sample(1:ncol(posterior_pd_pre), 100)
posterior_pd_pre_sample <- t(posterior_pd_pre)[sample_rows,]
ppc_dens_overlay(y = data_list_pre$y,
                 yrep = as.matrix(posterior_pd_pre_sample))

pre_loo <- pre_intervention_fit$loo(save_psis=TRUE)
plot(pre_loo)

# post-intervention code

data_list_post <- list(N=length(post_intervention_dat$anxiety_after),
                       y=post_intervention_dat$anxiety_after,
                       preint_score = post_intervention_dat$anxiety_before,
                       J_maj = post_intervention_dat$major%>%n_distinct(),
                       J_gender=post_intervention_dat$gender%>%n_distinct(),
                       gender = post_intervention_dat$gender,
                       major = post_intervention_dat$major,
                       upper = 50,
                       lower = 10,
                       mu_sigma=0,
                       tau_sigma=6,
                       mu_intercept = 30,
                       tau_intercept = 2.5,
                       mu_tau_gender = 2.5,
                       tau_tau_gender = 0.5,
                       mu_tau_maj = 2.5,
                       tau_tau_maj = 0.5,
                       mu_beta_pre = 0,
                       tau_beta_pre = 0.175,
```

```
                        mu_beta_Z=0,
                        tau_beta_Z=4.375,
                        treatment_status = post_intervention_dat$Z,
                        only_prior=1)


post_intervention_mod <- cmdstan_model("posttreatment.stan", compile = TRUE)


data_list_post$only_prior = 1
post_intervention_prior_fit <- post_intervention_mod$sample(data_list_post,
                                                            seed = 365,
                                                            refresh = 0,
                                                            parallel_chains = 4,
                                                            adapt_delta = 0.99)
prior_post <- post_intervention_prior_fit$draws()

prior_pd_post <- tn_pred_distn(prior_post)

prior_pd_plots <- ppd_plot(prior_pd_post)

data_list_post$only_prior = 0
post_intervention_fit <- post_intervention_mod$sample(data_list_post,
                                                      seed = 365,
                                                      refresh = 500,
                                                      parallel_chains = 4)
posterior_post<-post_intervention_fit$draws()

posterior_pd_post <- tn_pred_distn(posterior_post)

posterior_pd_plots <- ppd_plot(posterior_pd_post)

intercept_comparison <- plot_prior_posterior(prior_post,
                                             posterior_post, "beta_0")
tau_maj_comparison <- plot_prior_posterior(prior_post,
                                           posterior_post, "tau_maj")
tau_gender_comparison <- plot_prior_posterior(prior_post,
                                              posterior_post, "tau_gender")
sigma_comparison <- plot_prior_posterior(prior_post,
                                         posterior_post, "sigma")
mu_comparison <- plot_prior_posterior(prior_post,
                                      posterior_post, "mu")
beta_Z_comparison <- plot_prior_posterior(prior_post,
                                          posterior_post, "beta_Z")
beta_pre_comparison <- plot_prior_posterior(prior_post,
                                            posterior_post, "beta_pre")


grid.arrange(intercept_comparison, sigma_comparison)
grid.arrange(tau_maj_comparison, tau_gender_comparison)
grid.arrange(beta_Z_comparison, beta_pre_comparison)

mu_comparison
```

```
data_list_post$only_prior = 0
post_intervention_fit <- post_intervention_mod$sample(data_list_post,
                                                      seed = 365,
                                                      refresh = 500,
                                                      parallel_chains = 4)
posterior_post<-post_intervention_fit$draws()

posterior_pd_post <- tn_pred_distn(posterior_post)

posterior_pd_plots <- ppd_plot(posterior_pd_post)

min_post <- ppc_stat(y = data_list_post$y,
                     yrep=t(posterior_pd_post), stat = "min")
max_post <- ppc_stat(y = data_list_post$y,
                     yrep=t(posterior_pd_post), stat = "max")
skewness_post <- ppc_stat(y = data_list_post$y,
                          yrep= t(posterior_pd_post), stat = "skewness")
grid.arrange(min_post, max_post, ncol = 2)

skewness_post

sample_rows <- sample(1:ncol(posterior_pd_post), 100)
posterior_pd_post_sample <- t(posterior_pd_post)[sample_rows,]
ppc_dens_overlay(y = data_list_pre$y,
                 yrep = as.matrix(posterior_pd_post_sample))

post_loo <- post_intervention_fit$loo(save_psis=TRUE)
#print(post_loo)
plot(post_loo)


# draw 20 samples for each population data point (major/gender combo)
# rows of the returned matrix are in order of those in the population df
# note that the returned matrix only includes combinations of maj/gender that exist in
# the experimental data
estimate_pre_popn <- function(joined_df, popn_df, ppd_pre, n_samples = 20){
  #browser()
  pre_treatment_samples <- list()
  sampling_indexes <- list()
  #print(paste("nrows: ", nrow(popn_df)))

  # get 20 scores for each population data point
  for(row in 1:nrow(popn_df)){
    # get matching experimental row indices for each popn data point
    # from joined_df
    popn_sample <- popn_df[row, ]
    #print(paste("row: ", row, " popn_sample: ", popn_sample))
    exp_row_idxs <- joined_df %>%
      filter(gender == popn_sample$gender
             & major == popn_sample$major) %>%
      select(exp_idx) %>%
      pull(exp_idx) %>%
      unique()
```

```
    #print(paste("row ", row,
    #            " has matching indexes of length: " ,
    #           length(exp_row_idxs)))


    # subset the ppd_pre matrix with corresponding rows
    # iff we are able to
    if(length(exp_row_idxs) > 0){
      ppd_subset <- ppd_pre[exp_row_idxs, ]
      # sample from resulting subset
      pre_treatment_sample <- sample(ppd_subset, n_samples)
      #print(class(post_treatment_sample))
      pre_treatment_samples <- pre_treatment_samples %>%
        list.append(pre_treatment_sample)
      # append the index of the first element of the exp_row_idxs
      # corresponds to the row in the experimental dataframe
      # with this specific major/gender combo
      sampling_indexes <- sampling_indexes %>% list.append(exp_row_idxs[1])
    }
    #print(post_treatment_sample)
  }
  return(list(pre_treatment_samples, sampling_indexes))
}
matched_rows <- match_rows(exp_dat, pop_dat)
ret_list <- estimate_pre_popn(matched_rows,
                              pop_dat,
                              posterior_pd_pre)
pre_treatment_population <- ret_list[[1]]
subset_idxs <- ret_list[[2]]
pre_treatment_population <- as.matrix(pre_treatment_population)

pre_treat_popn_plot <- melt(ret_list[[1]]) %>% ggplot(aes(x = value)) +
        geom_histogram(alpha=1,
                       fill=color_scheme_get()$light,
                       color=color_scheme_get()$light_highlight)+
        labs(title="Pre-treatment Maths Anxiety Score - Population Distribution", x="Maths Anxiety Score
        theme_minimal()
pre_treat_popn_plot


# do multiple imputation for post-treatment using pre intervention scores
# draw 20 samples for each population data point (major/gender combo)
estimate_post_popn <- function(joined_df, majors, genders,pop_ids, pre_int_popn,
                               post_draws,
                               n_samples = 20){

  #browser()
  ate_samples <- matrix(nrow = nrow(pre_int_popn), ncol= n_samples)
  treatment_samples <-  matrix(nrow = nrow(pre_int_popn), ncol= n_samples)
  placebo_samples  <- matrix(nrow = nrow(pre_int_popn), ncol= n_samples)
  #print(paste("nrows: ", nrow(popn_df)))

  beta_0 <- subset_draws_df(post_draws, c("beta_0"))
```

```
beta_pre <- subset_draws_df(post_draws, c("beta_pre"))
beta_Z <- subset_draws_df(post_draws, c("beta_Z"))
sigma <- subset_draws_df(post_draws, c("sigma"))
u_major1 <- subset_draws_df(post_draws, c("u_major[1]"))
u_major2 <- subset_draws_df(post_draws, c("u_major[2]"))
u_major3 <- subset_draws_df(post_draws, c("u_major[3]"))
u_major4 <- subset_draws_df(post_draws, c("u_major[4]"))
u_major5 <- subset_draws_df(post_draws, c("u_major[5]"))
u_gender1 <- subset_draws_df(post_draws, c("u_gender[1]"))
u_gender2 <- subset_draws_df(post_draws, c("u_gender[2]"))
u_gender3 <- subset_draws_df(post_draws, c("u_gender[3]"))

random_effects_df <- data.frame(u_major1, u_major2,u_major3, u_major4,
                                u_major5, u_gender1, u_gender2,
                                u_gender3)

# loop over the members of population we were able to predict for
for(row in 1:nrow(pre_int_popn)){

  # find these parameters (major/gender)
  # should only be one combination
  # the set of all correct major, gender, Z combos
  # for each entry in column
  major <- majors[row,1]
  gender <- genders[row, 1]
  len <- n_samples


  # use the betas from the post-intervention model
  # and the predicted population pre-intervention scores
  # as covariates
  # recall that our model is:
  # y = beta_0 + beta_pre * pre_intervention + beta_Z * Z
  # + u_major + u_gender

  placebo <-  DescTools::Sample(beta_0, len) +
      DescTools::Sample(beta_pre, len) * pre_int_popn[row, ] +
      subset_helper(random_effects_df, "u_major", major, len) +
      subset_helper(random_effects_df, "u_gender", gender, len)

  treatment <- placebo +
      DescTools::Sample(beta_Z, len) * 1

  placebo <- placebo %>% pull(beta_0)
  treatment <- treatment %>% pull(beta_0)

  ate_samples[row, ] <- treatment-placebo
  treatment_samples[row, ] <- treatment
  placebo_samples[row, ] <- placebo

}

return(list(ate_samples, treatment_samples, placebo_samples))
```

```
}

# n_samples corresponds to the nrows of the population
subset_helper <- function(draws_df, name, value, n_samples=300){
  pasted <-  paste(name,".",value,".", sep="")
  subset <- draws_df %>% select(pasted)
  #subset <- subset_draws_df(draws_df,pasted)
  return(DescTools::Sample(subset, n_samples))
}




ret_list <- estimate_post_popn(matched_rows,
                               subset_idxs,
                               pre_treatment_population,
                               posterior_pd_post)
ate_distribution <- ret_list[[1]]
treatment_distribution <- ret_list[[2]]
placebo_distribution <- ret_list[[3]]


ate_plot <- melt(ate_distribution) %>% ggplot(aes(x = value)) +
        geom_histogram(alpha=1,
                       fill=color_scheme_get()$light,
                       color=color_scheme_get()$light_highlight)+
        labs(title="Average Treatment Effect", x="Average Treatment Effect")+
        theme_minimal()
treatment_distribution_plot <- melt(treatment_distribution) %>% ggplot()+ aes_string("value") +
        geom_histogram(alpha=1,
                       fill=color_scheme_get()$light,
                       color=color_scheme_get()$light_highlight)+
        labs(title="Population Post Intervention Score (Treatment)", x = "Post Intervention Score")+
        theme_minimal()

placebo_distribution_plot <- melt(placebo_distribution, "score") %>% ggplot()+ aes_string("value") +
        geom_histogram(alpha=1,
                       fill=color_scheme_get()$light,
                       color=color_scheme_get()$light_highlight)+
        labs(title="Population Post Intervention Score (Placebo)", x="Post Intervention Score")+
        theme_minimal()

ate_plot
```

Stan code

Pre-intervention score

```
functions{
  // from the stan docs
real normal_lub_rng(real mu, real sigma, real lb, real ub) {
  real p_lb = normal_lcdf(lb| mu, sigma);
  real p_ub = normal_lcdf(ub| mu, sigma);
  real u = uniform_rng(exp(p_lb), exp(p_ub));
```

```
  real y = mu + sigma * inv_Phi(u);
  return y;
}

}


data {
  int<lower=0> N; // num obs
  vector[N] y; // the data
  int<lower = 0> J_maj; // major categories
  int<lower = 0> J_gender; // gender categories
  int<lower = 1, upper = J_gender>  gender[N];
  int<lower = 1, upper = J_maj> major[N];

  //trunc norm bounds
  int upper;
  int lower;

  // prior inputs
  real mu_sigma;
  real tau_sigma;

  real mu_intercept;
  real tau_intercept;

  real mu_tau_gender;
  real tau_tau_gender;

  real mu_tau_maj;
  real tau_tau_maj;

  real only_prior;

}

parameters {
  real beta_0; // always have an intercept

  // random effect params
  real<lower = 0> tau_maj;
  vector<multiplier = tau_maj>[J_maj] u_major;

  real<lower = 0> tau_gender;
  vector<multiplier = tau_gender>[J_gender] u_gender;

  real<lower=0> sigma; // natural data variance
}

transformed parameters{
  vector[N] mu = beta_0 + u_gender[gender] + u_major[major];

}
```

```
model {
  //priors
  sigma ~ normal(mu_sigma, tau_sigma);
  beta_0 ~ normal(mu_intercept, tau_intercept);
  tau_gender ~ normal(mu_tau_gender,tau_tau_gender);
  tau_maj ~ normal(mu_tau_maj, tau_tau_maj);

  // random effects
  u_major ~ normal (0, tau_maj);
  u_gender ~ normal (0, tau_gender);

  //likelihood
  if(only_prior == 0){
    // y is trunc norm distributed
    target+= normal_lpdf(y | mu, sigma) -
            log_diff_exp(normal_lcdf(upper | mu, sigma),
                         normal_lcdf(lower | mu, sigma));
  }

}

generated quantities{
  vector[N] log_lik;
  //vector[N] y_pred;
  for (i in 1:N) {
    log_lik[i] = normal_lpdf(y[i] | mu[i], sigma) -
            log_diff_exp(normal_lcdf(upper | mu[i], sigma),
                         normal_lcdf(lower | mu[i], sigma));
  }

}
```

Post-intervention score

```
functions{
  // from the stan docs
real normal_lub_rng(real mu, real sigma, real lb, real ub) {
  real p_lb = normal_cdf(lb, mu, sigma);
  real p_ub = normal_cdf(ub, mu, sigma);
  real u = uniform_rng(p_lb, p_ub);
  real y = mu + sigma * inv_Phi(u);
  return y;
}

}


data {
  int<lower=0> N; // num obs
  vector[N] y; // the data
  int<lower = 0> J_maj; // major categories
  int<lower = 0> J_gender; // gender categories
  int<lower = 1, upper = J_gender>  gender[N];
  int<lower = 1, upper = J_maj> major[N];
```

```
  vector[N] treatment_status; // Z which is either 0 or 1
  vector[N] preint_score;

  //trunc norm bounds
  int upper;
  int lower;

  // prior inputs
  real mu_sigma;
  real tau_sigma;

  real mu_intercept;
  real tau_intercept;

  real mu_beta_pre;
  real tau_beta_pre;

  real mu_beta_Z;
  real tau_beta_Z;

  real mu_tau_gender;
  real tau_tau_gender;

  real mu_tau_maj;
  real tau_tau_maj;


  real only_prior;


}

parameters {
  real beta_0; // always have an intercept
  real beta_pre;
  real beta_Z;

  // random effect params
  real<lower = 0> tau_maj;
  vector<multiplier = tau_maj>[J_maj] u_major;

  real<lower = 0> tau_gender;
  vector<multiplier = tau_gender>[J_gender] u_gender;

  real<lower=0> sigma; // natural data variance
}

transformed parameters{
  vector[N] mu = beta_0 + u_gender[gender] + u_major[major] +
                 beta_Z*treatment_status + beta_pre*preint_score;

}

model {
```

```
  //priors
  sigma ~ normal(mu_sigma, tau_sigma);
  beta_0 ~ normal(mu_intercept, tau_intercept);
  tau_gender ~ normal(mu_tau_gender,tau_tau_gender);
  tau_maj ~ normal(mu_tau_maj, tau_tau_maj);
  beta_pre ~ normal(mu_beta_pre,tau_beta_pre);
  beta_Z ~ normal(mu_beta_Z,tau_beta_Z);

  // random effects
  u_major ~ normal (0, tau_maj);
  u_gender ~ normal (0, tau_gender);

  //likelihood
  if(only_prior == 0){
    // y is trunc norm distributed
    target+= normal_lpdf(y | mu, sigma) -
            log_diff_exp(normal_lcdf(upper | mu, sigma),
                         normal_lcdf(lower | mu, sigma));
  }

}

generated quantities{
  vector[N] log_lik;
  //vector[N] y_pred;
  for (i in 1:N) {
    log_lik[i] = normal_lpdf(y[i] | mu[i], sigma) -
            log_diff_exp(normal_lcdf(upper | mu[i], sigma),
                         normal_lcdf(lower | mu[i], sigma));
  }

}
```