STA365: Homework 1

Eric Zhu

05/02/2021

Contents

Quick Introduction

To begin, we will define two models with respect to two likelihoods:

1. The first model (model 1) will be based on this likelihood: $y_i \sim N(\exp(\alpha + \beta t_i), \sigma^2)$
2. The second model (model 2) will be based on this likelihood: $\log(y_i) \sim N(\alpha + \beta t_i, \sigma^2)$

Our data looks like (for visulization purposes only!):

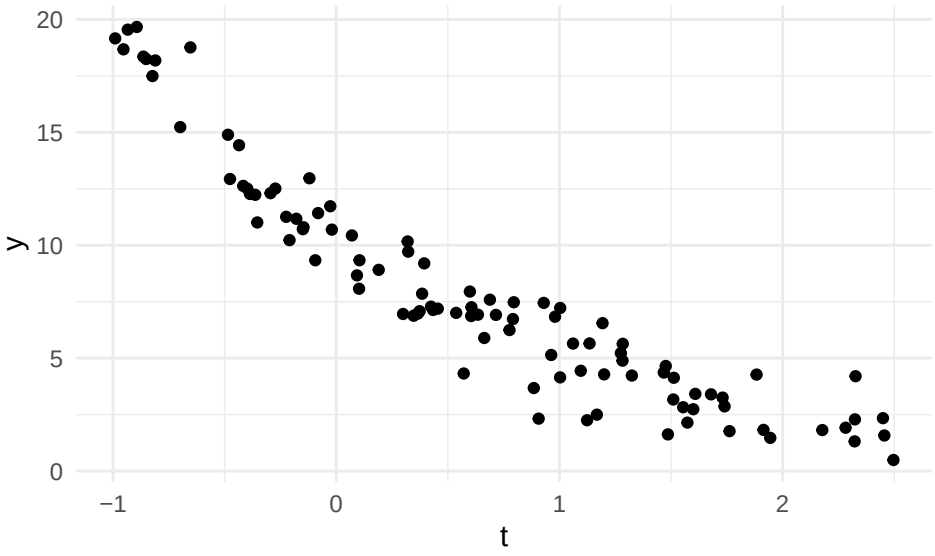

Figure 1: Scatterplot of y on t

Part 1: Priors

Overview

We know some information about y, that is:

1. It is not physically impossible to get $y < 0$, but should be fairly unlikely.
2. y and t should be negatively correlated.
3. It is very unlikely for us to see a value for y greater than 50.
4. It is always true that $-1 \leq t \leq 2.5$.

From both of the likelihoods, notice that both model will require 3 priors for these parameters: α, β, σ. We will use the normal distribution for all 3 parameters it is sensible to assume by the CLT that with large n, we would have approximately normal distributions. So for each prior we will specify μ, τ. Note that since there cannot be negative standard deviation/variance, σ is actually a half normal distribution. Also in the report, the the normal distribution will be parameterized with μ and σ rather than μ and σ^2 because normal distributions in Stan use μ and σ.

We don't know much about these priors given that the 4 bullet points we know are not super informative. But, we know a few crucial pieces of information: the range of max and min values for y, i.e.,y is unlikely to be greater than 50 and y is unlikely to be less than 0. Since we also know that $-1 \leq t \leq 2.5$, we can figure out some sensible priors by looking the extremes of t. Note that since we are using normal (or half normal) distributions for priors, we will use 2τ as a sensible cutoff for determining our prior parameters because calculating $\mu \pm 2\tau$ allows us to capture most of the density of the normal distribution ($\approx 95\%$ of values).

Prior for model 1

To begin we will try putting a prior on α, β. Note that since we have $\mu_i = \exp(\alpha + \beta t_i)$, we will need to be careful with how large we make the means of these two priors.

As stated in the quick introduction, we will being by examining y values at the extreme values of t. We know that since y, t are negatively correlated, we have $\mu_i = \exp(\alpha + \beta t_i)$ to be monotonically decreasing, so our maximum values of y occur when t is close to -1 and our minimums occur when t is close to 2.5.

So thinking about our constraints when $t = -1$, we realize that we want $\exp(\alpha + \beta \cdot -1) \leq 50$, and when $t = 2.5$, we realize that we want $\exp(\alpha + \beta \cdot 2.5) \geq 0$. As such we can form a equation using our two constraints:

$$\frac{\exp(\alpha + \beta \cdot -1)}{\exp(\alpha + \beta \cdot 2.5)} = \frac{45}{0.5}$$

Note that I've chosen 0.5 to equal $\exp(\alpha + \beta \cdot 2.5)$ and 45 to equal $\exp(\alpha + \beta \cdot -1)$ because it should be unlikely for $y < 0$ and $y > 50$, and we need to factor in the spread of y_i and the spread of the prior distributions.

From our relation above, we can solve for β using the exponent rules to get:

$$\begin{aligned} \frac{\exp(\alpha)\exp(\beta \cdot -1)}{\exp(\alpha)\exp(\beta \cdot 2.5)} &= \frac{45}{0.5} \\ \frac{-\beta}{2.5\beta} &= \frac{45}{0.5} \end{aligned}$$

Thus, solving for β, we get -1.28566, which is negative and conforms to the y, t are negatively correlated constraint, and should allow us to stay within our constraints regarding likely values of y.

To find α, we need to solve $\exp(\alpha + \beta \cdot -1) = 45$. Since we have $\beta = -1.28566$, we can calculate a value for α, which comes out to 2.521.

These values for α, β represent one set of rather "extreme" possible values for α, β because we have set $\exp(\alpha + \beta \cdot 2.5) = 0.5$, and $\exp(\alpha + \beta \cdot -1) = 45$, and 0.5 and 45 represent very nearly improbable values of y. Additionally, $\beta = -1.28566$ is an extreme one since it represents a steep slope with a corresponding intercept that is decidedly large while also constrained on $-1 \leq t \leq 2.5$.

On the flip slide, we can have the other extremes of α and β by using the following equation:

$$\frac{\exp(\alpha + \beta \cdot -1)}{\exp(\alpha + \beta \cdot 2.5)} = \frac{10}{0.5}$$

We choose the values 10 and 0.5 because we aren't given much information about y such as it's PDF, but we do know that it is unlikely to be less than 0 and greater than 50. Since we had already gotten β, α values for a steep slope and large intercept, we wanted to get values for a slope that is relatively close to 0 (while still having a negative correlation between y, t) and an intercept that is relatively small.

Solving for α, β using the same procedure we did above we get: $\alpha = 1.44666, \beta = -0.855924$. We now have two values each for α, β, one for each extreme. If we suppose that each of these points correspond to either $\mu \pm 2\tau$, e.g., -1.28566 $= \mu_\beta - 2\tau$ and $-0.855924 = \mu_\beta + 2\tau_\beta$, we can get the parameters for prior distributions of α, β. To find μ_β we will take the simple average of the two values of β, i.e., $\frac{-1.28566 - 0.855924}{2} = -1.07079$. Since we assumed that these two points corresponded to $\mu_\beta \pm 2\tau_\beta$, we get $\tau_\beta = 0.107434$ with $\mu_\beta = -1.07079$. We can do the same operations to find μ_α, τ_α, which come out as: $\mu_\alpha = 1.98383, \tau_\alpha = 0.268585$.

So to recap, we now have values the parameters for the distributions of $\alpha \sim N(\mu_\alpha, \tau_\alpha), \beta \sim N(\mu_\beta, \tau_\beta)$:

$$\alpha \sim N(1.98383, 0.268585), \beta \sim N(-1.07079, 0.107434)$$

To find parameters for σ, we will begin with μ_σ. We hope that our mean for y_i is close or on the data points, so we will set $\mu_\sigma = 0$. Now recall that $0 \leq y \leq 50$, so we can write the following equation that will help set τ_σ:

$$\frac{\exp((\mu_\alpha + 2\tau_\alpha) + (\mu_\beta - 2\tau_\beta) \cdot -1) + (\mu_\sigma + 2\tau_\sigma)}{\exp((\mu_\alpha - 2\tau_\alpha) + (\mu_\beta - 2\tau_\beta) \cdot 2.5) - (\mu_\sigma + 2\tau_\sigma)} = \frac{50}{0.05}$$

This equation allows us to find what value of τ_σ we need such that $0 \leq y \leq 50$ when we plug in values for α, β, σ that are at approximately the 95^{th} percentile of their respective prior distributions. The numerator of this equation has the highest possible intercept, steepest possible slope, and most additive noise we can have while limited to our $\pm 2\tau$ "cutoff". We should find it extremely unlikely to see y values greater than 50 or less than 0 where $-1 \leq t \leq 2.5$.

Plugging in -1.28566 for $\mu_\beta - 2\tau_\beta$, 2.521 for $\mu_\alpha + 2\tau_\alpha$, 1.44666 for $\mu_\alpha - 2\tau_\alpha$ and 0 for μ_σ, we can solve for τ_σ, which we get as: 0.069245.

To recap, we get the following distributions for our priors:

1. $\alpha \sim N(1.98383, 0.268585)$
2. $\beta \sim N(-1.07079, 0.107434)$
3. $\sigma \sim N_+(0, 0.069245)$

Discussion of the Priors: why we have weakly informative priors and prior checks

In constructing the priors, we incorporated all 4 bullet points. Off the bat, knowing that y, t were negatively correlated was crucial for knowing that β needed a distribution where the probability density was entirely (or almost entirely) over negative values. Next, the two bullet points detailing the most likely values of y were considered the most since they most explicitly provided the range of likely values for y, which allowed us to decide on where to centre the means of our prior distributions and also decide on how spread out likely values for our priors should be. Additionally, knowing that $-1 \leq t \leq 2.5$, allowed us to realize on such a relatively small interval, a steeper slope would result in much more extreme values at $t = -1$ and $t = 2.5$. Combining all this information allowed us to go through the process of hypothesizing values for α and β such that using values in the tails of the two distributions in the equation $\exp(\alpha + \beta t)$ would still result in values where $0 \leq \exp(\alpha + \beta t) = y \leq 50$. Keep in mind that we also knew that we needed to put a distribution on σ, so when setting the parameters for α, β, we made sure to choose "safe" parameters such that when adding likely values for σ to $\exp(\alpha + \beta t)$, we would still mostly generate values of y above 0 and under 50.

While these 4 pieces of information was substantial, we still have weakly informative priors. First we have weakly informative priors because we know the scale of the data; it is given to us that $-1 \leq t \leq 2.5$ and y is likely between 0 and 50. Another reason is that after drawing from the prior predictive distribution, we see that while almost all of the density is concentrated within the range of likely y values, i.e., below 50 and greater than 0, we also see density above 50, which means that unlikely but possible y values still occur under our prior. We can see this from the plot of prior predictive distributions:

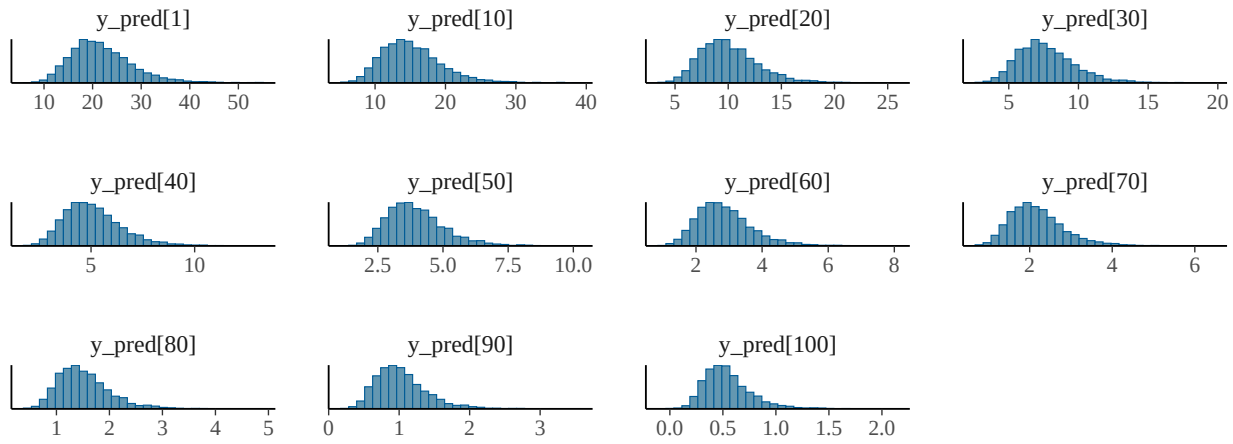

Figure 2: Draws from prior predictive distribution, top left is distribution of draws with smallest t, bottom right is distribution of draws with largest t - model 1

In particular, we see that in the plot of the distribution with the smallest t (the one in the top left above) there is still density very close to 50, which should be decidedly unlikely. We also see that in the plot of `y_pred[100]`, there are negative values, which should also be "fairly unlikely". However, we aren't given access to how fast the density decreases near 50, so we don't know in mathematical terms how unlikely seeing $y > 50$ actually is. In contrast, knowing how heavy the tails are would allow us to construct our priors such that our tails better represent plausible data set rather than covering these more implausible data set. This uncertainty does give evidence to our argument that we have weakly informative priors.

We can also examine the histograms of the draws from the prior μ values from the prior:

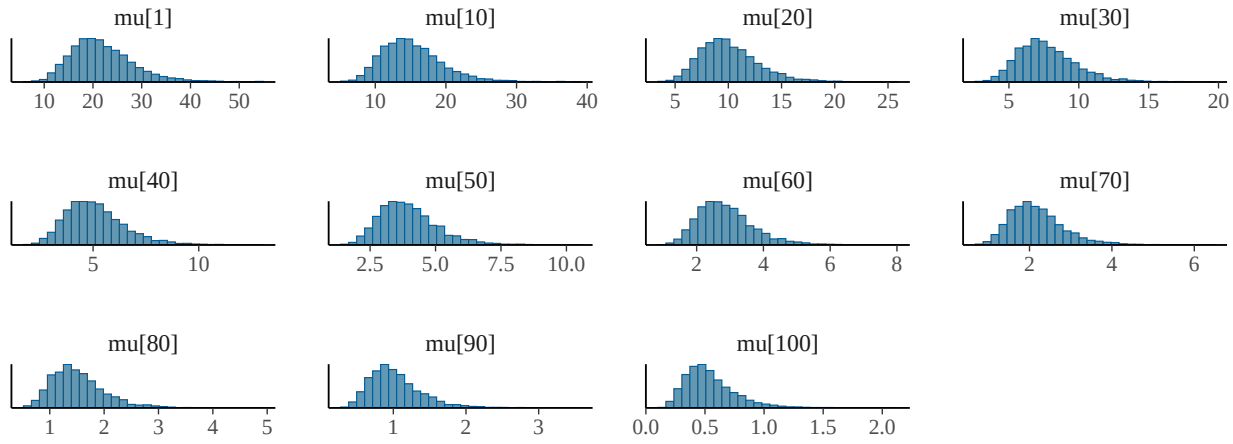

Figure 3: Draws of mu from the prior, top left is distribution of draws with smallest t, bottom right is distribution of draws with largest t - model 1

We also see that as t increases towards 2.5, the centre of the distribution of μ also moves towards 0 as we would expect. Here we also similarly to the plot of `y_pred[1]` that `mu[1]` also has density close to or above 50, and on the other extreme, we see that the plot of `mu[100]` has a left tail very close to 0. Note here that we won't get values below 0 because the range of the $exp()$ function is strictly non-negative. So again, we have evidence that we have weakly informative priors as the distributions for μ are able to cover unlikely values of y.

Additionally, we see that as t increases, the corresponding distributions move fairly fast to the left, indicating that our distribution is not too constrained as we can see that our `mu[100]` distribution does not cover values such as 40 to 50. Since we also see the same behaviour with the `y_pred` distributions, we have evidence that these priors are sensible.

Finally, it is worth mentioning that there's an enlarged side by side plot of the `mu[1]` and `y_pred[1]` distributions in the appendix under "Model 1 Prior Plots". These can allow us to better examine the extreme values in the tails of the respective distributions.

Prior for model 2

To begin, we recall that the likelihood defines the $log(y)$ to be normally distributed. As such, we will first preprocess the data by applying a log transformation. There's a reference plot of the log-transformed data in the appendix under "Model 2 prior plots".

We will be following a very similar set of processes in determining the priors for this model as we did with the first model. Note that we have to consider our y values in terms of $log(y)$, but since the log transformation is monotonic, a lot of our analysis from model 1 holds in this section.

As stated in the quick introduction, we will being by examining y values at the extreme values of t. We know that since y, t are negatively correlated, we have $\mu_i = \alpha + \beta t_i$ to be monotonically decreasing, so our maximum values of y occur when t is close to -1 and our minimums occur when t is close to 2.5.

So thinking about our constraints when $t = -1$, we realize that we want $\alpha + \beta \cdot -1 \leq log(50)$, and when $t = 2.5$, we realize that we want $\alpha + \beta \cdot 2.5 \geq log(0)$. Unlike the first model, we will use a system of equations as we can't apply exponent rules:

$$\begin{cases} \alpha + \beta \cdot -1 = log(45) \\ \alpha + \beta \cdot 2.5 = log(0.5) \end{cases}$$

Note that we've chosen $log(45)$ to equal $\alpha + \beta \cdot -1$ because it should be unlikely for $y > 50$, and we've chosen $\alpha + \beta \cdot 2.5 = log(0.5)$ because it is close to 0 and we avoid the problem with $log(0)$ being undefined.

Solving the system of equation we get:

$$\begin{cases} \alpha = 2.521 \\ \beta = -1.28566 \end{cases}$$

We note that our system of equations here and our equation in the model 1 section will result in equivalent values for α, β given that we fix the same y values at $t = -1, t = 2.5$. More formally for some $x, y \in \mathbb{R}$,

$$\begin{cases} \alpha + \beta \cdot -1 = log(x) \\ \alpha + \beta \cdot 2.5 = log(y) \end{cases}$$
$$\iff$$
$$\frac{\exp(\alpha + \beta \cdot -1)}{\exp(\alpha + \beta \cdot 2.5)} = \frac{x}{y}$$

So next, we calculate:

$$\begin{cases} \alpha + \beta \cdot -1 = log(10) \\ \alpha + \beta \cdot 2.5 = log(0.5) \end{cases}$$

Solving for α, β we get: $\alpha = 1.44666, \beta = -0.855924$. We now have two values each for α, β, one for each extreme as we did in model 1. To recap, the first set of α, β are those with a very steep slope and large intercept (as we are constrained on $t \in [-1, 2.5]$), and the second set are those with a slope close to 0 and a small intercept. As we did with model 1, if we suppose our values are either $\mu \pm 2\tau$, e.g., $-1.28566 = \mu_\beta - 2\tau$ and $-0.855924 = \mu_\beta + 2\tau_\beta$, we can get the parameters for prior distributions of α, β. To find μ_β we will take the simple average of the two values of β, i.e., $\frac{-1.28566 - 0.855924}{2} = -1.07079$. Since we assumed that these two points corresponded to $\mu_\beta \pm 2\tau_\beta$, we get $\tau_\beta = 0.107434$ with $\mu_\beta = -1.07079$. We can do the same operations to find μ_α, τ_α, which come out as: $\mu_\alpha = 1.98383, \tau_\alpha = 0.268585$.

We now have values the parameters for the distributions of $\alpha \sim N(\mu_\alpha, \tau_\alpha), \beta \sim N(\mu_\beta, \tau_\beta)$:

$$\alpha \sim N(1.98383, 0.268585), \beta \sim N(-1.07079, 0.107434)$$

To find parameters for σ, we will begin with μ_σ. We again hope that our mean for y_i is close or on the data points, so we will set $\mu_\sigma = 0$. Now recall that $0 \leq y \leq 50$, so we can write the following equation that will help set τ_σ:

$$(\mu_\alpha + 2\tau_\alpha) + (\mu_\beta - 2\tau_\beta) \cdot -1 + (\mu_\sigma + 2\tau_\sigma) = log(50)$$

The equation will allow us to find a value of τ_σ that is a "conservative" choice as we plug in values for α, β, σ that are at approximately the 95^{th} percentile of their respective prior distributions. In other words $(\mu_\alpha + 2\tau_\alpha)$ will be a high intercept, $\mu_\beta - 2\tau_\beta$ will be a steep slope, and so the value for τ_σ should be small enough that $y \geq log(50)$ should still be very rare. Note that since we need to undo the log scale after drawing from the predictive distribution (using $\exp(log(y))$), we will always have non-negative y values in the end, so our information that $y < 0$ is unlikely isn't as much as a consideration as it was model 1.

Plugging in -1.28566 for $\mu_\beta - 2\tau_\beta$, 2.521 for $\mu_\alpha + 2\tau_\alpha$, 1.44666 for $\mu_\alpha - 2\tau_\alpha$ and 0 for μ_σ, we can solve for τ_σ, which we get as: 0.052682

To recap, we get the following distributions for our priors:

1. $\alpha \sim N(1.98383, 0.268585)$
2. $\beta \sim N(-1.07079, 0.107434)$
3. $\sigma \sim N_+(0, 0.052682)$

Discussion of the Priors: why we have weakly informative priors and prior checks

In constructing the priors, we again incorporated all 4 bullet points. In fact, our process of constructing the priors was almost exactly the same as model 1, except on a log scale; the following paragraphs will be similar the analysis of model 1. However, a key difference is that μ will not be constrained to non-negative numbers as μ is no longer the result of exp().

As with model 1, knowing y, t were negatively correlated was crucial for knowing that β needed a distribution where the probability density was entirely (or almost entirely) over negative values. Next, the two bullet points detailing the most likely values of y were considered the most since they most explicitly provided the range of likely values for y, which allowed us to decide on where to centre the means of our prior distributions and also decide on how spread out likely values for our priors should be. Additionally, knowing that $-1 \leq t \leq 2.5$, allowed us to realize on such a relatively small interval, a steeper slope would result in much more extreme values at $t = -1$ and $t = 2.5$. Combining all this information allowed us to go through the process of hypothesizing values for α and β such that using values in the tails of the two distributions in the equation $\alpha + \beta t$ would still result in values such that $0 \leq \alpha + \beta t = \exp(log(y)) \leq 50$. Keep in mind that we also knew that we needed to put a distribution on σ, so we when setting the parameters for α, β, we made sure to choose "safe" parameters such that when adding likely values for σ to $\alpha + \beta t$, we would still mostly generate values of y under 50. We did not care about values under 0 because we would undo the log transformation after drawing from the predictive distribution.

We again have weakly informative priors because we know the scale of the data; it is given to us that $-1 \leq t \leq 2.5$ and y is likely between 0 and 50 as our data hasn't changed. Another reason is that after transforming draws from the prior predictive distribution back into our data's orginal scale, we see that while almost all of the density is concentrated within the range of plausible y values, (i.e., below 50 and greater than 0) we also see density above 50, meaning that unlikely but possible y values still occur under our prior.

We can see this in the `mcmc_hist` plot below:

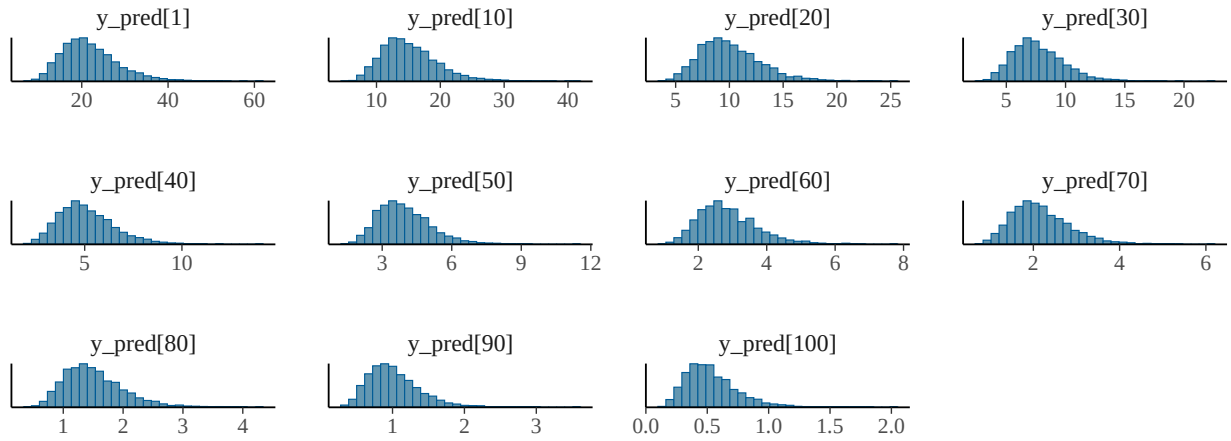

Figure 4: Draws from prior predictive distribution, top left is distribution of draws with smallest t, bottom right is distribution of draws with largest t - model 2

In particular, we see that in the plot of the distribution with the smallest t (the one in the top left above) there is still density above 50 (even around 60), which should be very implausible y values. Of course, we make such a statement under a decent amount of uncertainty because we aren't given access to how fast the density decreases near 50, even with a log transformation. In contrast, knowing how heavy the tails are would allow us to construct our priors such that our tails better represent plausible data set rather than covering these more implausible data set. This uncertainty does give evidence to our argument that we have weakly informative priors.

It is worth noting with model 2, we see more density above 50 when comparing the distributions of `y_pred[1]` with that of model 1. We also see slightly longer right tails on all of the distributions above, but between the two models, the centres of the distribution seem to move towards 0 at a similar rate as t increases to 2.5.

We can also examine the histograms of the draws from the posterior μ values from the posterior:

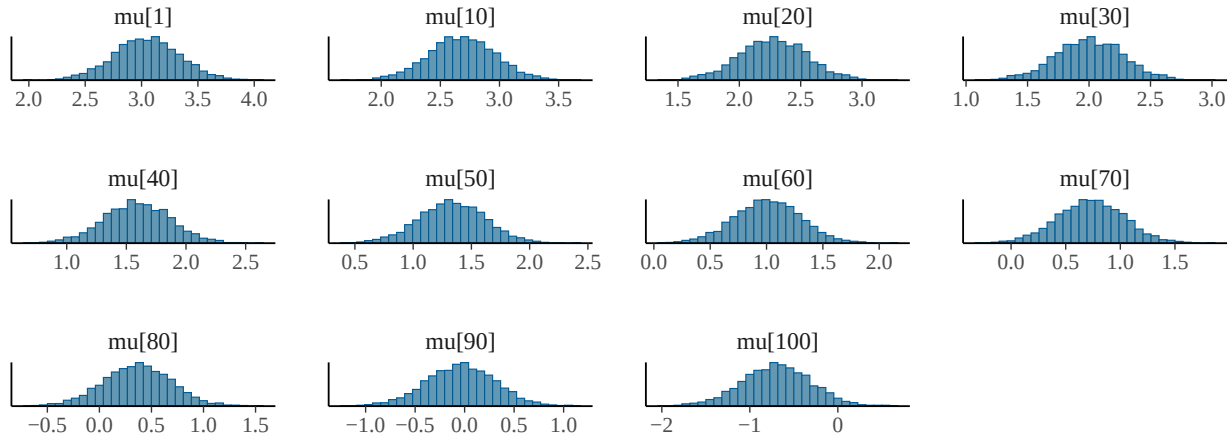

Figure 5: Draws of mu from the prior, top left is distribution of draws with smallest t, bottom right is distribution of draws with largest t - model 2

We also see that as t increases towards 2.5, the centre of the distribution of μ also moves towards 0 as we would expect, although unlike model 1, the center of `mu[100]` actually goes below 0 given that μ is no longer the result of the exp() function. Since the distribution of `mu[1]` shows values above 4, we find that we should expect to find values around $\exp(4) \approx 54.5$ if we applied the exp() function to every value in the distribution

of `mu[1]`. So again, we have evidence that we have weakly informative priors as the distributions for μ are able to cover unlikely values of y.

Additionally, we see that as t increases, the corresponding distributions move at a sensible rate to the left, indicating that our distribution is not too constrained as we can see that our `mu[100]` distribution does not cover values above 4, i.e., values in the right tail of `mu[1]`. Also, the range of values for μ we get on this log-scale are not implausible (given our information about y) when we convert it back into our original scale by applying the exp() function. Since we also see the same behaviour with the `y_pred` distributions, we have evidence that these priors are sensible.

It is worth mentioning again that there's an enlarged side by side plot of the `mu[1]` and `y_pred[1]` distributions in the appendix under "Model 2 Prior Plots". These can allow us to better examine the extreme values in the tails of the respective distributions. Please note that the distribution of `mu[1]` is in the log scale.

Part 2: Posteriors

Individual critiques of models

Model 1

To begin our model critique, we first examine the distributions of the μ, α parameters drawn from the posterior distribution vs the prior distribution:

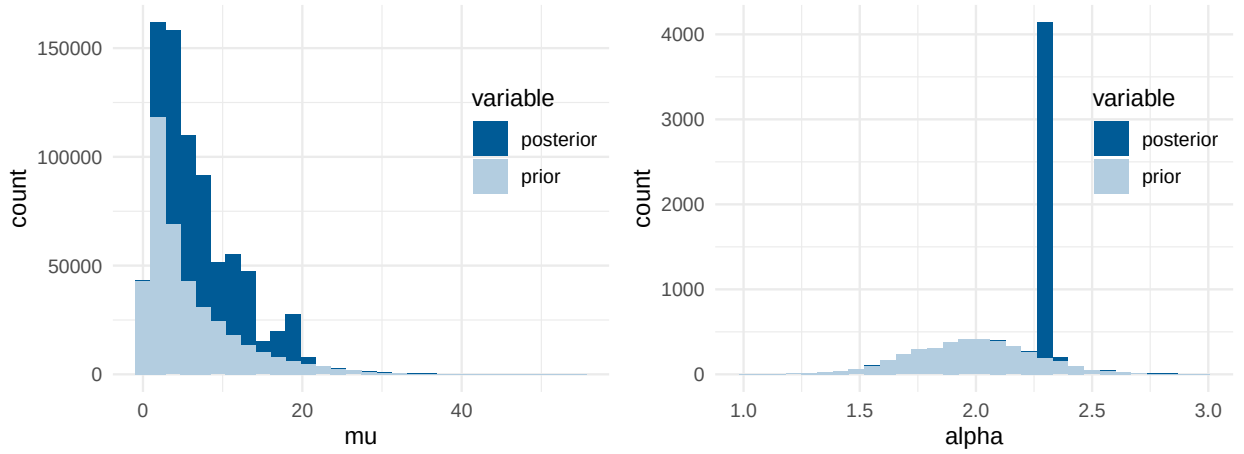

Figure 6: Draws of mu and alpha from the posterior vs the prior - model 1

From the two plots above, we see behaviour we expect to see with weakly informative prior; the posterior distribution has less spread and stays within the prior, which shows the regularizing properties of the weakly informative priors. So we clearly don't have much evidence if any of prior-data conflict as our scales on our prior distribution are sensible, we haven't grossly misunderstood the information we were given about the dataset. However, this changes a bit when we look at the comparisons between the distributions of draws of σ and β from the prior and posterior distributions:

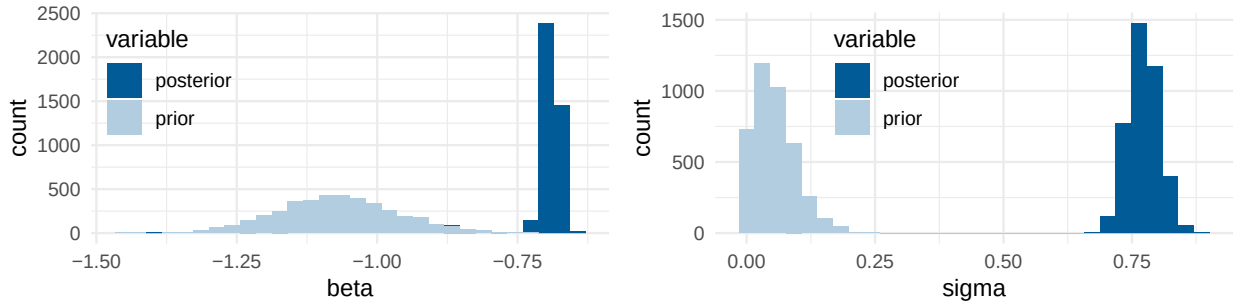

Figure 7: Draws of beta and sigma from the posterior vs the prior - model 1

We see that plot regarding β has the distribution of β values from the posterior overlapping the very end of the right tail of the values from the prior. However, the prior has a really long tail and the posterior is much "skinner" than the prior. So perhaps we have some tweaking to do with respect to the β prior, but this behaviour is very similar to the example of the cauchy prior we looked at in class. However, the behaviour of σ shows evidence of prior-data conflict as the distribution of σ values from the posterior are far away from those of the prior, and worse, the distribution of values from the posterior is about as spread as the prior. This may be a problem relating to us having very little information regarding the likelihood of $y > 50$ or $y < 0$ since that information factored into our thinking for the σ prior.

Next, to evaluate our fit, we can evaluate test statistics plots of min, max, and skewness, which are all ancillary test statistics given our normal distribution (they aren't highly dependent on the distribution parameters).

Looking at the min, max test statistic plots, we see:

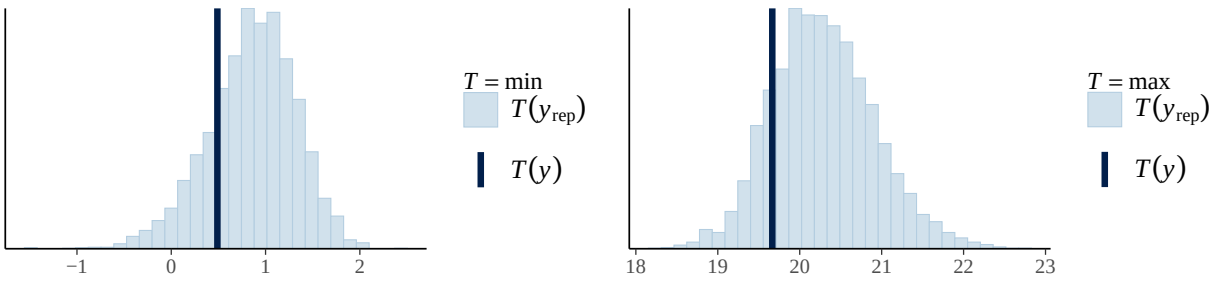

Figure 8: Test statistics plots of min, max - model 1

We gather that the model was able to fairly well capture the min and max of the data as the $T(y)$ line in both plots are close to the centre of the $T(y_{rep})$ distribution. Our observation also holds with the skewness test statistic plot:

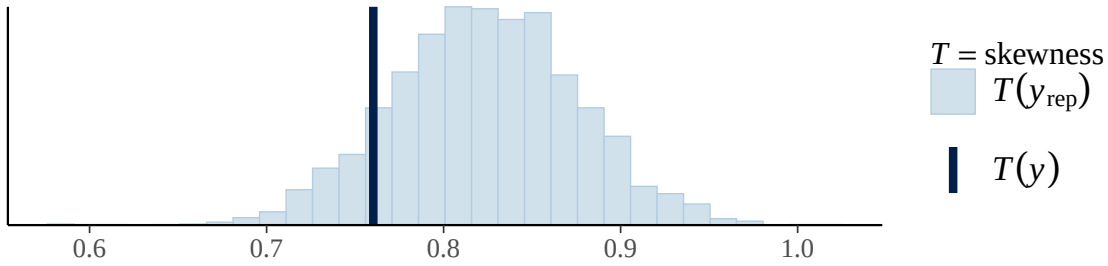

Figure 9: Test statistics plots of sknewness - model 1

From examining these 3 plots, we see that the model performed well in capturing the 3 test statistics of our observed data, indicating that our model fit the data well.

Next, comparing the standard deviation of the prior predictive distribution with that of the posterior predictive distribution, we see that the standard deviations from the posterior predictive distribution are on average far smaller. For example, `y_pred[1]` from the posterior predictive distribution had a standard deviation of ≈ 0.82 while `y_pred[1]` from the prior predictive distribution had a standard deviation of ≈ 6.37. However, if we examine `y_pred[100]` from the posterior predictive distribution, we find that its standard deviation is still ≈ 0.82, but the standard deviation of `y_pred[100]` from the prior predictive distribution we find that its standard deviation is ≈ 0.22. It seems a bit strange that the distribution from the prior predictive distribution is tighter than the distribution from the posterior predictive distribution; it could be indicative of the prior data conflict we saw from our plot of σ, where the centre of the σ distribution from the posterior had a far higher centre than that from the prior. It could also be due to the posterior μ distribution having it's density more spread out than the prior μ distribution, whose distribution (despite having a really long right tail) is incredibly concentrated around 0. A plot of distributions generated from the posterior predictive distribution can be found in the appendix under "Model 1 posterior plots" for reference.

Next, we'll examine the time course plot:

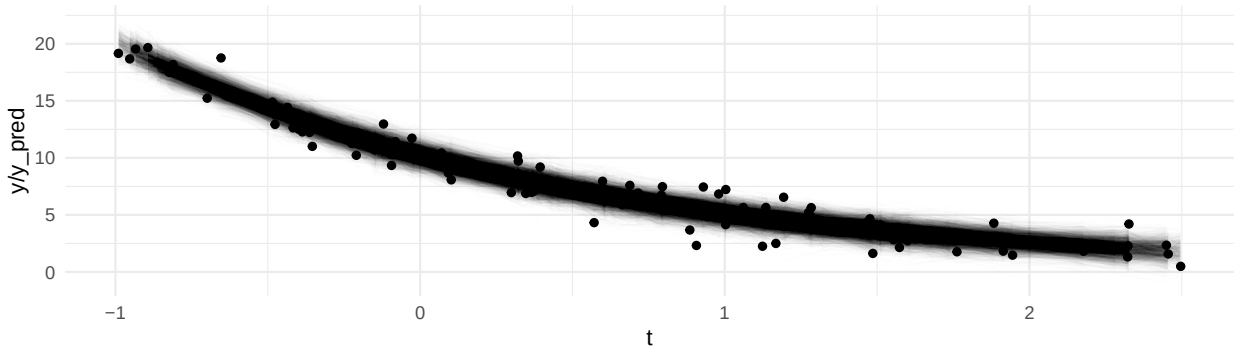

Figure 10: Time course plot - model 1

We can see that the model was able to generate data that looks very similar to the observed data. While it's clear from the graduated shading that many of the replicated data sets are concentrated around the posterior mean, some replicated data set were even able to capture some noisier data points. For example, there are some observed data points around $t = 2$ that deviate somewhat from the rest of the points in the same region, but the model was able to generate data that also capture those data points. A good example of this is observed data point (2.32670030, 4.2000518), as some generated data points at $t = 2.32$ were close to 4.2. We can also quickly examine the `ppc_dens_overlay` plot to see how the density estimate of the observed data compares to that of our replicated data sets:

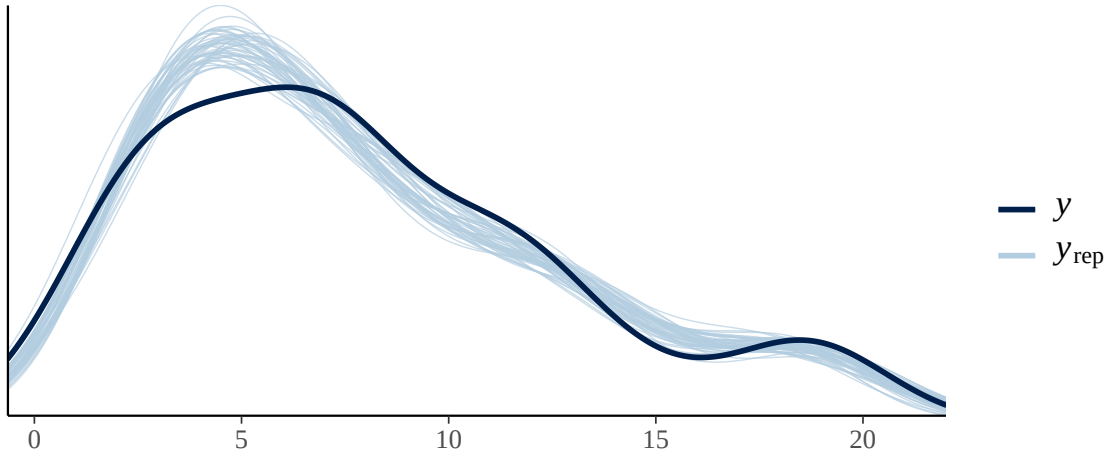

Figure 11: Estimated density of y vs replicated dataset density - model 1

From the plot we see that the model performed poorly in replicating the density of y values around 5, i.e., the replicated data sets had a higher density of values around 5 than the observed one. There also appears to be less density around values of 10 than the observed data set and values around 20, but this was far better in comparison to values around 5. The model appears to perform fairly well on both tails, even capturing the dip in density of y values around 15. Overall, the density of replicated data sets are similar to those of the observed data set for the majority of y values, giving us evidence that our model is a good fit for the data.

While this seems to be a sign of good model performance, we should be concerned with possible overfitting as being able to generate predictions close to every observed data point is suspect. As such, we have plotted the PSIS plot generated from our leave one out cross validation:

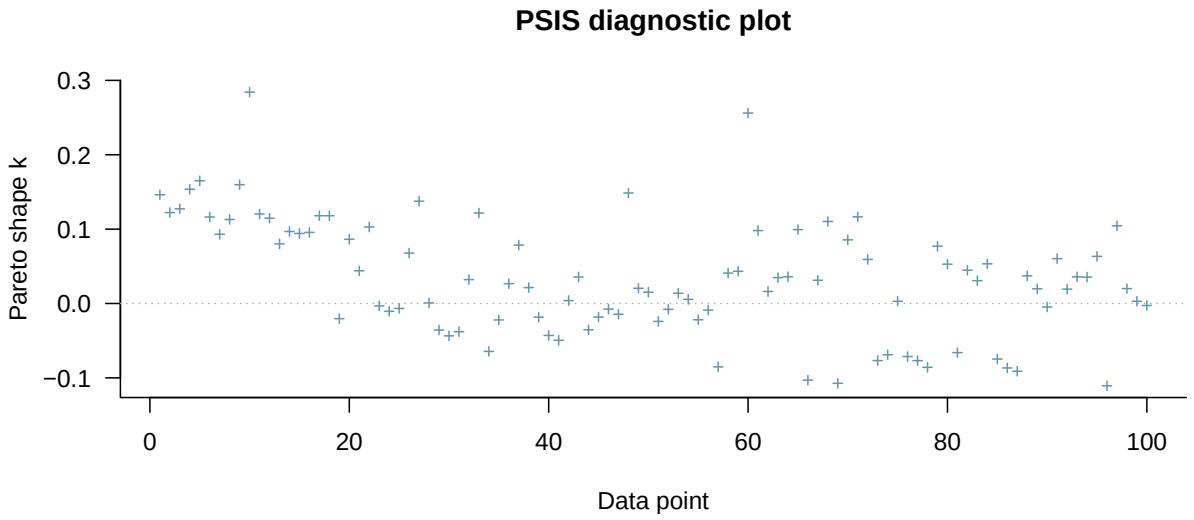

Figure 12: PSIS Plot - model 1

From the PSIS plot, we gather that our $\hat{k}$ values are good for all data points, using the $\hat{k} < 0.5$ rule. So it seems that we have a well fit model for our observed data set, and we should not be too concerned about overfitting. In fact, our highest $\hat{k}$ value is only about 0.3, and most of the $\hat{k}$ values fall within the range of -0.1 to 0.2, so we conclude that our LOO predictive distributions are similar to our full data predictive distribution (no influential points). Considering the rest of our analysis in this section, e.g., test statistic plots and the `ppc_dens_overlay` plot, we have substantial evidence that our model is a good fit (and appropriate) for the data.

Model 2

We will again begin our model critique by examining the distributions of the μ, α parameters drawn from the posterior distribution vs the prior distribution:

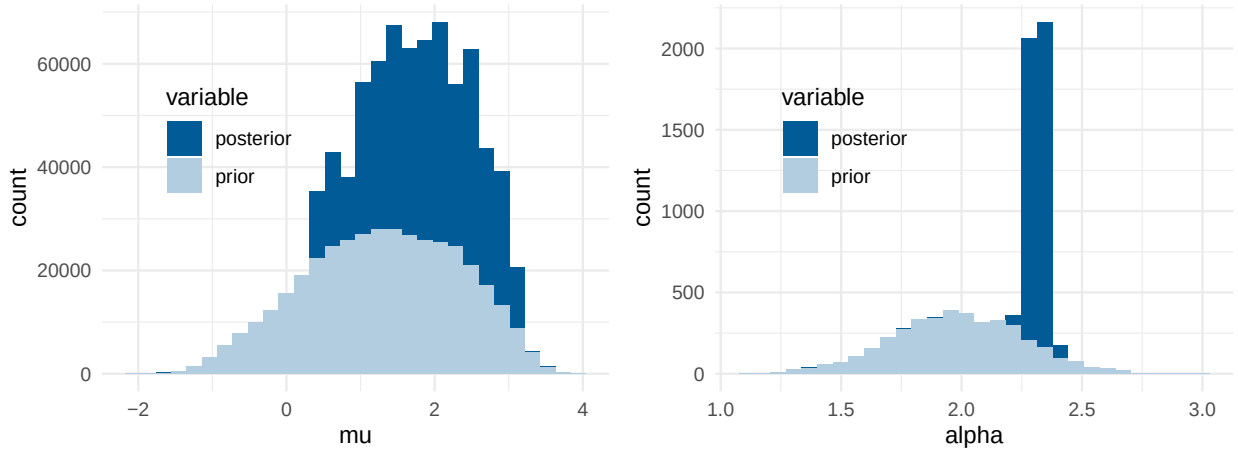

Figure 13: Draws of mu and alpha from the posterior vs the prior - model 2

From the two plots above, we again see behaviour we expect to see with weakly informative prior; the posterior distribution has decidedly less spread and stays within the prior. Like model 1, we have evidence that our priors have a regularizing effect. Like model 1, so far we don't have much evidence if any of prior-data conflict as our scales on our prior distribution are sensible, and we haven't grossly misunderstood the information we were given about the data set. This story again changes when we look at the comparisons between the distributions of draws of σ and β from the prior and posterior distributions:

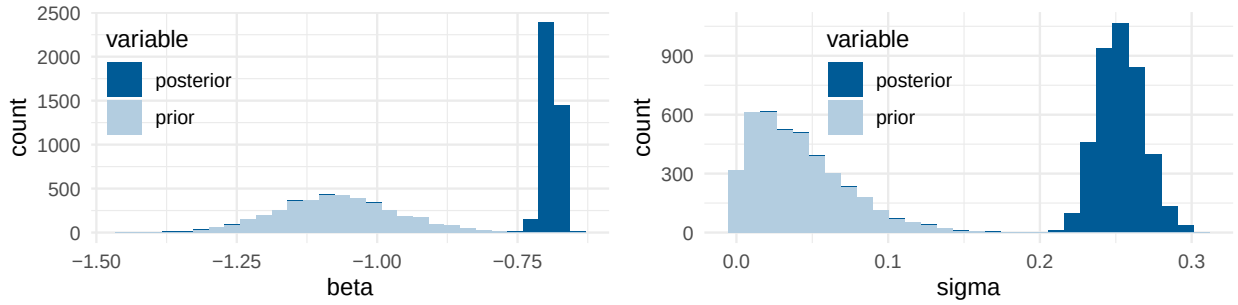

Figure 14: Draws of beta and sigma from the posterior vs the prior - model 2

We see that plot of β values has the distribution of posterior β values overlapping the very end of the right tail of the values from the prior very similarly to model 1. We may also want to tune the beta prior as suggested in model 1. The behaviour of σ shows evidence of prior-data conflict as the distribution of σ values from the posterior are far away from those of the prior (only a small amount of overlapping density between the right tail of the prior distribution and left tail of the posterior distribution), and the posterior distribution is only somewhat less spread out.

Next, to evaluate our fit, we can again evaluate test statistics plots of min, max, and skewness (they're ancillary as we have normal distributions for model 2).

Looking at the min, max test statistic plots, we see:

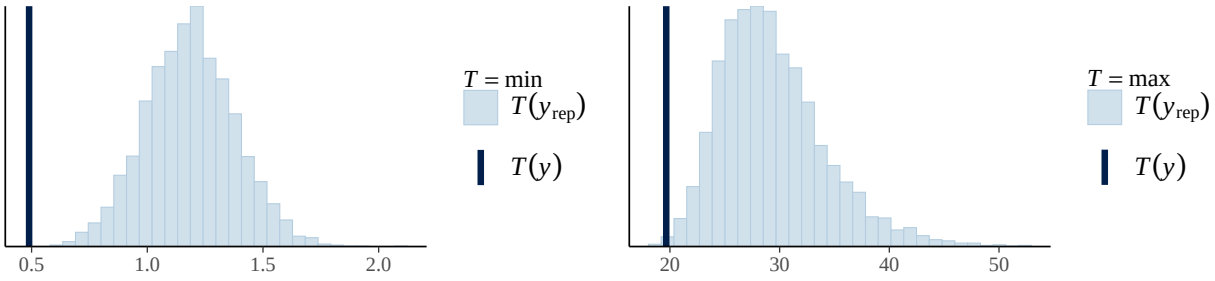

Figure 15: Test statistics plots of min, max - model 2

We gather that the model was somewhat able to capture the max of the observed data as $T(y)$ is in the left tail, while the model was unable to capture the min observed data. The skewness test statistic plot also shows that the model was somewhat able to capture the skewness of the observed data:

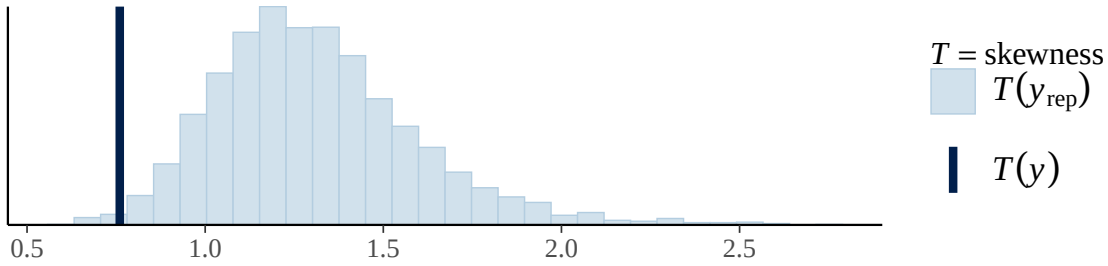

Figure 16: Test statistics plots of sknewness - model 2

It is worth noting that model 2 performed worse at capturing ancillary test statistics than model 1, and the model's failure to capture the min test statistic is a bit concerning.

Next, comparing the standard deviation of the prior predictive distribution with that of the posterior predictive distribution, we see that the standard deviations from the posterior predictive distributions are smaller than those from the prior predictive distribution, which is an indication of a good fit. This is in contrast to model 1, as with model 2 we also see that the posterior distributions of μ and σ are different. In particular, the posterior distribution of μ has a similar shape and centre to the prior distribution, and the posterior distribution of σ has some overlap with the prior. So in this respect, model 2 exhibits behaviour we expect from a well fit model in comparison to model 1. A plot of distributions generated from the posterior predictive distribution can be found in the appendix under "Model 2 posterior plots" for reference.

Next, we'll examine the time course plot:

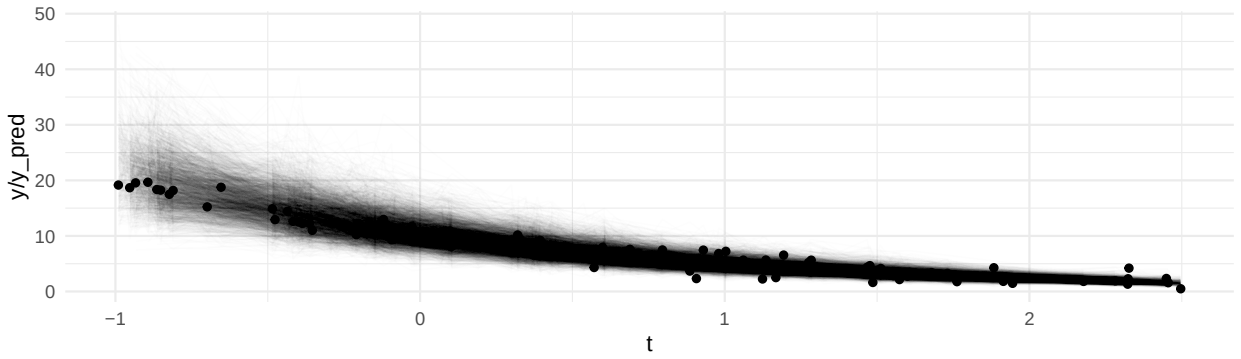

Figure 17: Time course plot - model 2

From the time course plot, we see that there is a lot of variation in the replicated data sets until around 0.5, where the replicated data sets get more concentrated around the mean from the posterior distribution. In fact, when $-1 \leq t < 0.5$ the replicated data sets have an incredibly wide spread with some replicated data sets predicting values of about 30 on the high end to 10 on the low end, which isn't totally unexpected since our prior predictive distribution generated improbable y values, e.g., $y > 50$. Additionally, it is encouraging to see from the shading that the majority of the replicated data sets are concentrated where there is a high density of observed data too; the model doesn't appear to be severely underfitting, i.e., biased.

We can also quickly examine the `ppc_dens_overlay` plot as we did with model 1 to see how the density estimate of the observed data compares to that of our replicated data sets:

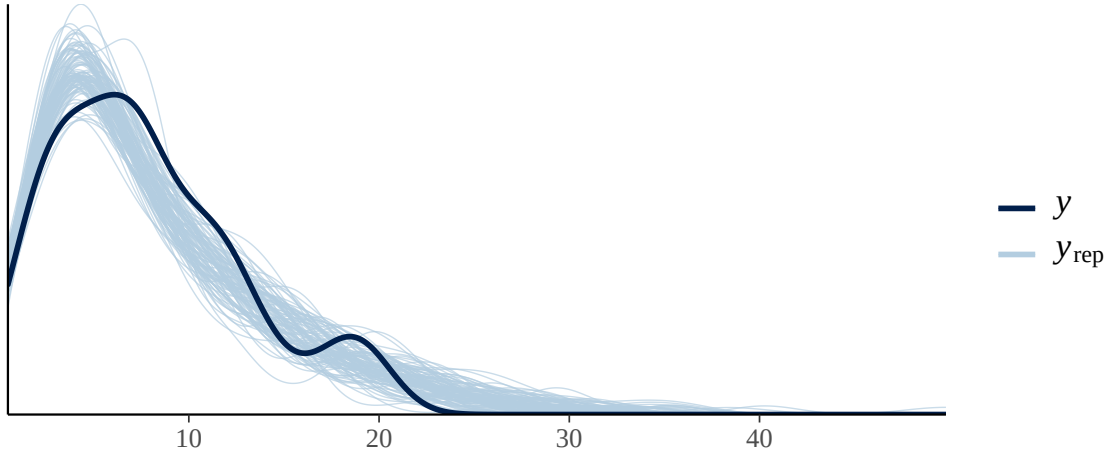

Figure 18: Estimated density of y vs replicated dataset density - model 2

Immediately, we see the large variation in the replicated data sets that we observed in the time course plot. It's especially notable that the replicated data sets had density in values near and above 30, which did not exist in the observed data set or model 1. Additionally, we see that this model does not capture the behaviour of the observed data density curve especially the "up-down dip" around y values of 15 very well. While some of the replicated data sets do appear to replicate that behaviour, most don't appear to; as a whole the replicated data sets show a pretty homogeneously smoothly decreasing right tail. Model 2 does seem to be able to replicate the peak in density of y values around 5 though. It appears that model 2 cannot capture some of the granularity in the data well given it's behaviour we've observed in the time course plot and in this plot. We'll now examine the PSIS plot to see how the model performed with cross validation and conclude if the model fit the data well and if it is an appropriate model:

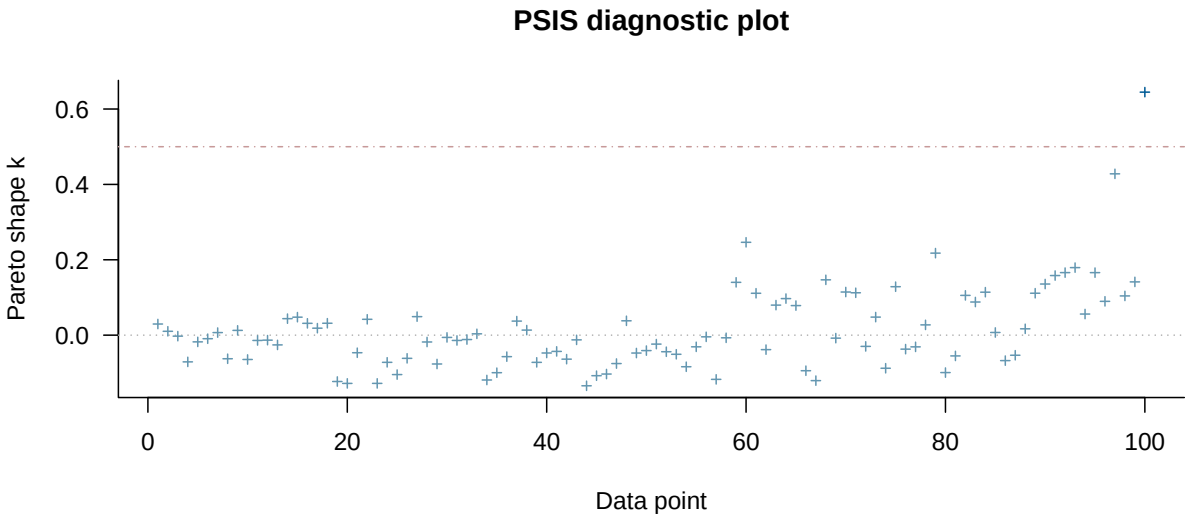

Figure 19: PSIS Plot - model 1

From our PSIS plot we see that there is one data point that has a particularly high $\hat{k}$ value of over 0.6. There is also one that has a $\hat{k}$ value that is close to 0.5 (≈ 0.45). The majority of the data points had $\hat{k}$ values of around -0.1 to 0.2 though. Between our one influential point and one nearly influential point, we have some evidence that this model was not a particularly good fit for the observed data, and perhaps it was not able to capture some of the granularity of the data, leading it to worse predictions through our cross validation procedure. Our results here are in stark contrast to model 1's PSIS plot, where the highest $\hat{k}$ value was around 0.3.

In the next section, we'll take our analysis of model 1 and model 2 and decide on which model was superior for our observed data set.

Comparison of models: picking the superior model

Quick note: while we've already touched base on how model compares in the individual critiques, we'll expand upon it here explicitly as it's more natural to do so in a comparison section rather than the individual critiques sections.

Since the prior vs posterior plots of the parameters for both models were relatively similar, we'll begin this section with the test statistic plots. From the min, max, skewness test statistics, we gather that both models were able to capture the test statistics. Model 2 was worse at doing so, especially when it came to the min test statistic, and it's worth noting that model 1's test statistic distributions had centres really close to the test statistics generated from the observed data. So from the test statistic plots, we have some evidence that model 1 was a more appropriate model.

Our haunch carries over to time course plots and density overlay plots. Starting with the time course plots, we see that model 1's replicated data sets were far less spread out than model 2. In fact, the concentration of the replicated data sets closely matched the concentration of the data points; it's pretty clear that model 1 was a good fit. While model 2 also seemed like a good fit for values where $t > 0.5$, there was a lot more variation in the replicated data sets where $t \leq 0.5$. This trend carried over to the density overlay plots as with model 1, the distribution of replicated data sets were both very tightly grouped and shaped, mostly resembling the density estimate of the observed data. Whereas with model 2, the distribution of replicated data sets were not very tightly grouped and had varying shapes, and did not really resemble the density estimate of the observed data, e.g., did not have a dip in density around y values of 15. From these two sets of plots, we have more evidence that model 1 was the superior model for this data set. Our PSIS plots

further confirmed this as model 2 had one influential point as determined by our $\hat{k}$ values, while model 1 did not have any, indicating once again that model 1 was a more appropriate model for our data.

Finally, we'll examine our ELPD values to further evaluate the predictive ability of our models:

Table 1: loo_compare results

	elpd_diff	se_diff	elpd_loo	se_elpd_loo	p_loo	se_p_loo	looic	se_looic
model1	0.00000	0.00000	-180.6318	15.89802	5.613154	1.163438	361.2637	31.79604
model2	-21.59582	14.79194	-202.2276	13.07534	5.916663	2.674633	404.4553	26.15068

Here we see that model 1 has a smaller (less negative) ELPD than model 2. Since ELPD can be considered similar to squared error (and scaled squared error in the case of a normal distribution), model 1 seems to have better predictive ability (-180.63 vs -202.23) as it could predict data closer to our observed data during cross validation. However, it is worth noting that the standard error of the difference is 14.79, so the difference in ELPD between the two models isn't that significant. Still, provided that model 2 had an influential point and generally higher $\hat{k}$ values (including our posterior predictive check plots, e.g., time course or test statistic plots), it seems that we have more evidence pointing to model 1 being a superior (and more appropriate) model for this data set.

Appendix

Additional plots

Model 1 prior plots

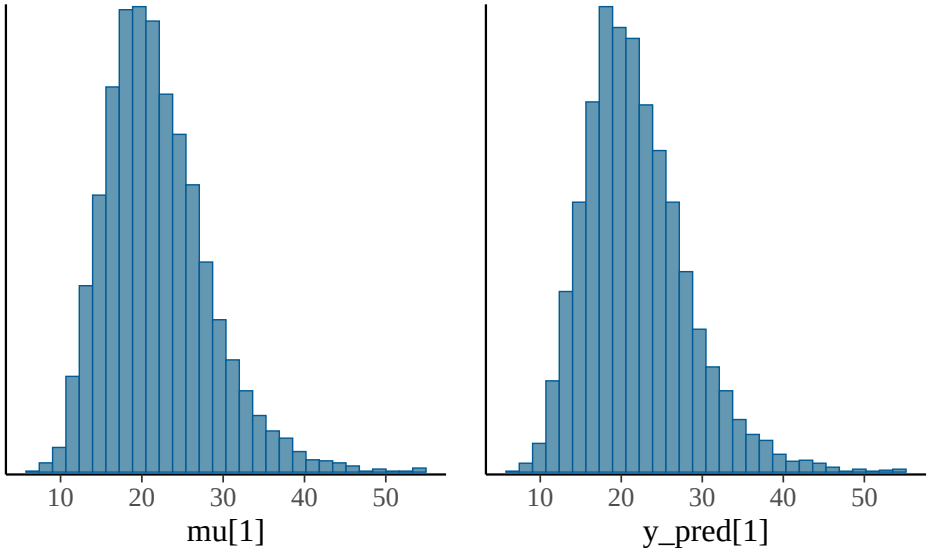

Figure 20: Histograms generated using the prior distribution with the smallest t - model 1

Larger sized plots of the distribution of `mu[1]` and `y_pred[1]`.

Model 1 posterior plots

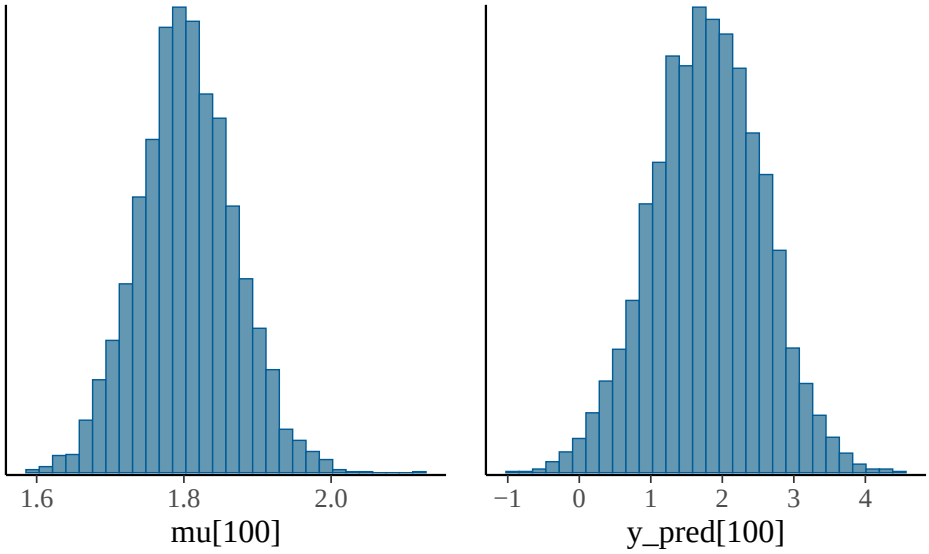

Figure 21: Histograms generated using the posterior distribution with the largest t - model 1

Larger sized plots of the distribution of `mu[100]` and `y_pred[100]`.

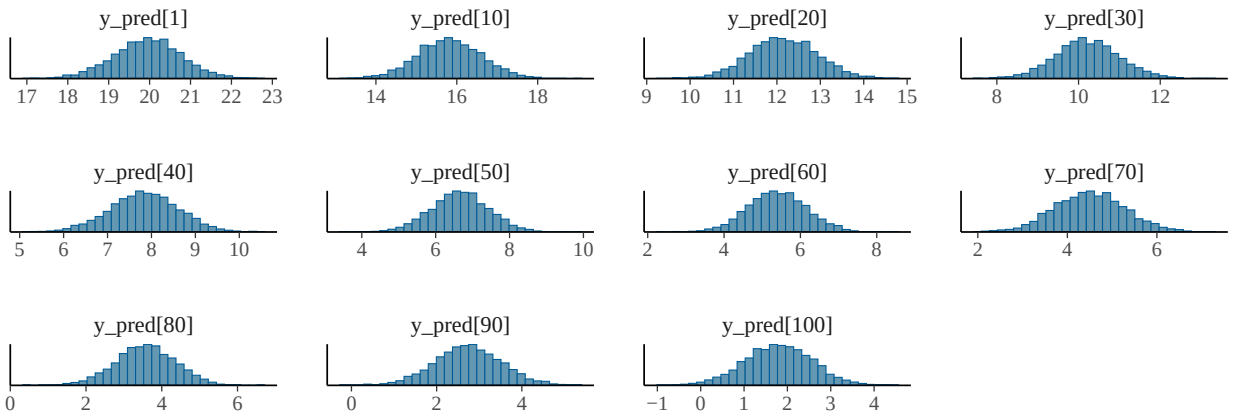

Figure 22: Distributions from posterior predictive distribution, top left corresponds to smallest t, bottom right corresponds to largest t - model 1

Model 2 prior plots

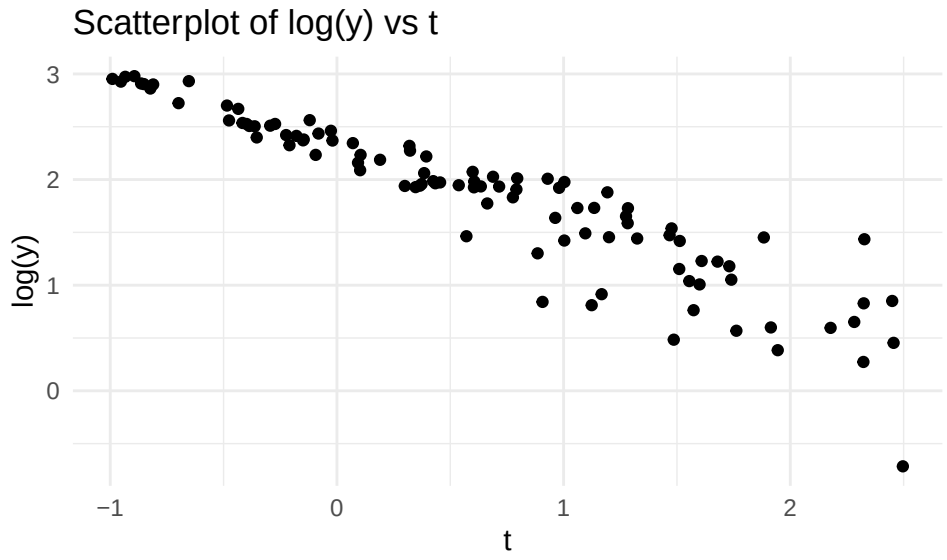

Figure 23: Scatterplot of log(y) on t - model 2

Note that the above plot should and will only be used for visualization purposes, not for determining priors!

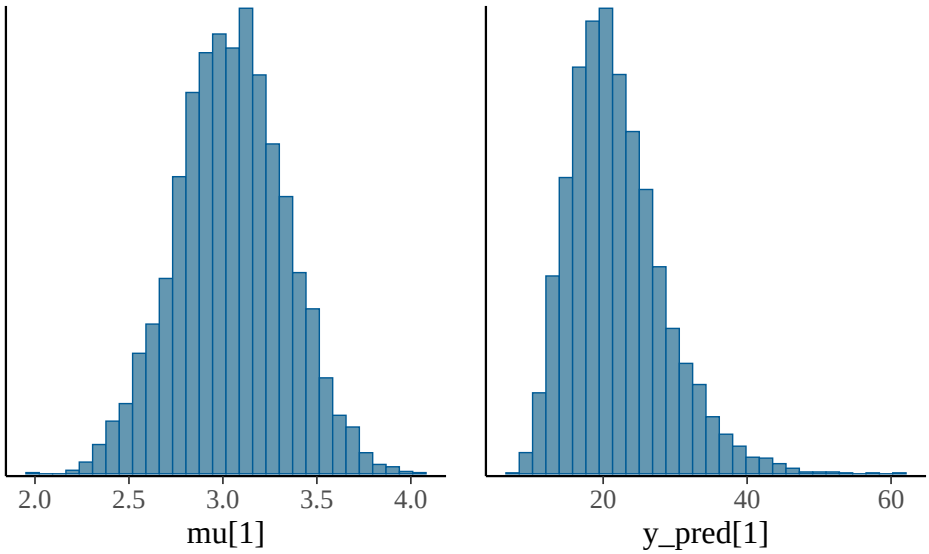

Figure 24: Histograms generated using the prior distribution with the smallest t - model 2

Larger sized plots of the distribution of `mu[1]` and `y_pred[1]`.

Model 2 posterior plots

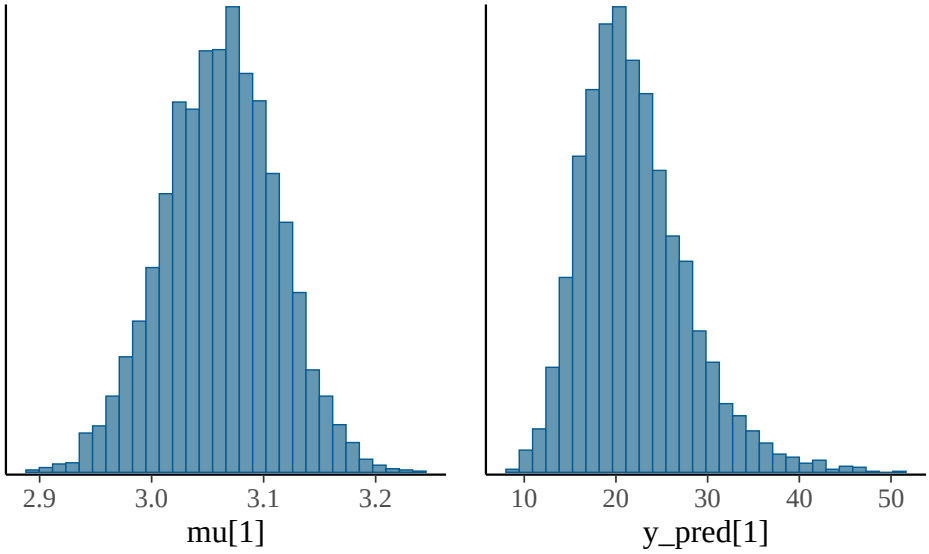

Figure 25: Histograms generated using the posterior distribution with the smallest t - model 2

Larger sized plots of the distribution of `mu[1]` and `y_pred[1]`.

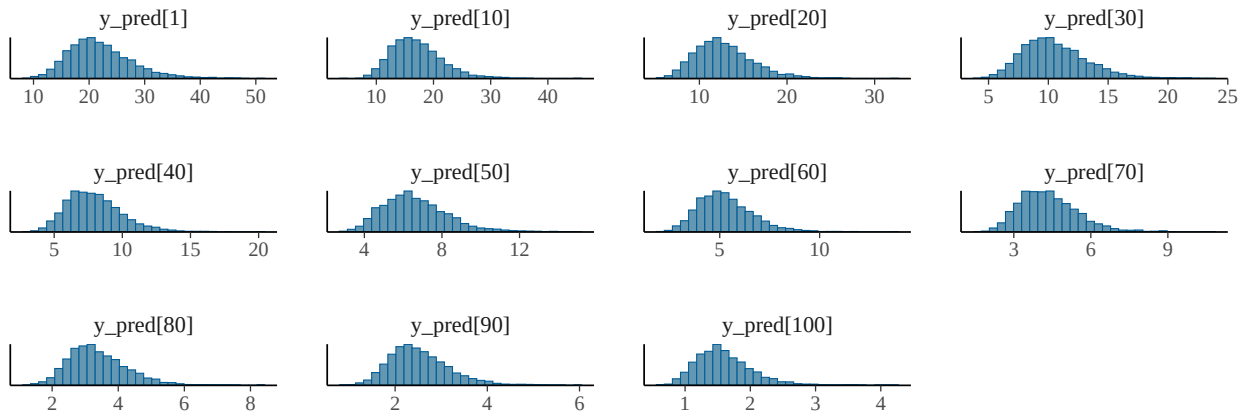

Figure 26: Distributions from posterior predictive distribution, top left corresponds to smallest t, bottom right corresponds to largest t - model 2

Code appendix

R Code

```
# library imports
library(cmdstanr)
library(loo)
library(tidyverse)
library(posterior)
library(bayesplot)
library(latex2exp)
library(reshape2)
library(gridExtra)
library(PerformanceAnalytics)

register_knitr_engine(override = TRUE)


data <- readRDS("hw1_data.RDS")
data <- data[order(data$t),]
data_list <- list(N=length(data$y),
                  y=data$y ,t = data$t) # model 1
data_list2 <- list(N=length(data$y),
                   y=log(data$y), t = data$t) # model 2

data %>% ggplot(aes(x=t, y=y)) +
  geom_point()+
  labs(x="t", y="y",
       title = "Scatterplot of y vs t")+theme_minimal()

mod1 <- cmdstan_model("model1.stan", compile = TRUE)


data_list$only_prior = 1

data_list$mu_alpha = 1.98383
```

```
data_list$mu_beta = -1.07079
data_list$mu_sigma = 0.0

data_list$tau_alpha = 0.268585
data_list$tau_beta = 0.107434
data_list$tau_sigma = 0.069245

prior_fit <- mod1$sample(data_list, seed = 365,refresh = 0)


prior_fit$summary()

mu_draws <- prior_fit$draws(c("mu[1]", "mu[10]", "mu[20]", "mu[30]", "mu[40]",
                              "mu[50]", "mu[60]", "mu[70]", "mu[80]",
                              "mu[90]", "mu[100]"))
pred_draws <- prior_fit$draws(c("y_pred[1]", "y_pred[10]",
                                "y_pred[20]", "y_pred[30]",
                                "y_pred[40]", "y_pred[50]",
                                "y_pred[60]", "y_pred[70]",
                                "y_pred[80]", "y_pred[90]",
                                "y_pred[100]"))

mod2 <- cmdstan_model("model2.stan", compile = TRUE)

data_list2$only_prior = 1

data_list2$mu_alpha = 1.98383
data_list2$mu_beta =  -1.07079
data_list2$mu_sigma = 0.0

data_list2$tau_alpha = 0.268585
data_list2$tau_beta = 0.107434
data_list2$tau_sigma = 0.052682

prior_fit2 <- mod2$sample(data_list2,seed = 365, refresh = 0)

mu_draws_m2 <- prior_fit2$draws(c("mu[1]","mu[10]",
                                  "mu[20]", "mu[30]", "mu[40]",
                                  "mu[50]", "mu[60]", "mu[70]",
                                  "mu[80]", "mu[90]", "mu[100]"))
pred_draws_m2 <- prior_fit2$draws(c("y_pred[1]", "y_pred[10]", "y_pred[20]",
                                    "y_pred[30]", "y_pred[40]", "y_pred[50]",
                                    "y_pred[60]", "y_pred[70]", "y_pred[80]",
                                    "y_pred[90]", "y_pred[100]"))


mcmc_hist(pred_draws)+scale_fill_manual(values=c(color_scheme_get()$light))
mcmc_hist(mu_draws)


mcmc_hist(pred_draws_m2)+scale_fill_manual(values=c(color_scheme_get()$light))
mcmc_hist(mu_draws_m2)
```

```
data_list$only_prior = 0

fit <- mod1$sample(data_list, seed = 365,refresh = 0)

#fit$summary()
mu_draws_post <- fit$draws(c("mu[1]", "mu[10]", "mu[20]", "mu[30]", "mu[40]",
                             "mu[50]", "mu[60]", "mu[70]",
                             "mu[80]", "mu[90]", "mu[100]"))
pred_draws_post <- fit$draws(c("y_pred[1]", "y_pred[10]", "y_pred[20]",
                               "y_pred[30]", "y_pred[40]", "y_pred[50]",
                               "y_pred[60]", "y_pred[70]", "y_pred[80]",
                               "y_pred[90]", "y_pred[100]"))
model1_loo <- fit$loo(save_psis=TRUE)
# class example code
ypreds_m1 <- fit$draws() %>% reshape2::melt() %>%
  filter(str_detect(variable, "y_pred") ) %>%
  extract(col = variable, into = "ind",
          regex = "y_pred\\[([0-9]*)\\]",
          convert = TRUE)
ypreds_m1 <- ypreds_m1 %>%
  mutate(time = data_list$t[ind],
         chain_iter = glue::glue("chain {chain}, iteration {iteration}"),
         .keep = "unused") %>%
  rename(y_pred = value)

mu_posterior<-melt(
  as_draws_matrix(subset_draws(fit$draws(), regex =TRUE, variable = "mu"))) %>%
  mutate(variable = str_replace_all(variable,
                                    pattern="mu.*",
                                    replacement = "posterior"))
mu_prior <- melt(as_draws_matrix(subset_draws(prior_fit$draws(), regex =TRUE,
                                              variable = "mu"))) %>%
  mutate(variable = str_replace_all(variable, pattern="mu.*", replacement = "prior"))

mu_comparison_df <- rbind(mu_prior, mu_posterior)

mu_plt <- ggplot(mu_comparison_df,aes(x=value, fill = variable)) +
  geom_histogram(alpha=1) +
  scale_fill_manual(
    values=c(color_scheme_get()$mid_highlight,
             color_scheme_get()$light_highlight)) +
  theme_minimal() +  theme(legend.position=c(.85, .65)) + labs(x="mu")



alpha_posterior<-melt(as_draws_matrix
                      (subset_draws(fit$draws(),
                                    regex =TRUE, variable = "alpha"))) %>%
  mutate(variable = str_replace_all(variable, pattern="alpha*",
                                    replacement = "posterior"))
alpha_prior <- melt(as_draws_matrix(subset_draws(prior_fit$draws(),
                                                 regex =TRUE,
                                                 variable = "alpha"))) %>%
```

```
  mutate(variable = str_replace_all(variable, pattern="alpha*",
                                    replacement = "prior"))

alpha_comparison_df <- rbind(alpha_posterior, alpha_prior)

alpha_plt <- ggplot(alpha_comparison_df,aes(x=value, fill = variable)) +
  geom_histogram(alpha=1) +
  scale_fill_manual(values=c(color_scheme_get()$mid_highlight,
                             color_scheme_get()$light_highlight)) +
  theme_minimal() +  theme(legend.position=c(.85, .65)) + labs(x="alpha")


beta_posterior<-melt(as_draws_matrix(
  subset_draws(fit$draws(), regex =TRUE,
               variable = "beta"))) %>%
  mutate(variable = str_replace_all(variable, pattern="beta*",
                                    replacement = "posterior"))
beta_prior <- melt(as_draws_matrix(
  subset_draws(prior_fit$draws(), regex =TRUE, variable = "beta"))) %>%
  mutate(variable = str_replace_all(variable, pattern="beta*",
                                    replacement = "prior"))

beta_comparison_df <- rbind(beta_posterior, beta_prior)

beta_plt <- ggplot(beta_comparison_df,aes(x=value, fill = variable))+
  geom_histogram(alpha=1) +
  scale_fill_manual(values=
                      c(color_scheme_get()$mid_highlight,
                        color_scheme_get()$light_highlight)) +
  theme_minimal() +  theme(legend.position=c(.15, .70))+ labs(x="beta")

grid.arrange(mu_plt, alpha_plt, ncol = 2)

sigma_posterior<-melt(as_draws_matrix(
  subset_draws(fit$draws(), regex =TRUE,
               variable = "sigma"))) %>%
  mutate(variable =
           str_replace_all(variable,
                           pattern="sigma*",
                           replacement = "posterior"))
sigma_prior <- melt(as_draws_matrix(subset_draws(
  prior_fit$draws(), regex =TRUE, variable = "sigma"))) %>%
  mutate(variable = str_replace_all(variable,
                                    pattern="sigma*",
                                    replacement = "prior"))

sigma_comparison_df <- rbind(sigma_posterior, sigma_prior)

sigma_plt <- ggplot(sigma_comparison_df,aes(x=value, fill = variable))+
  geom_histogram(alpha=1) +
  scale_fill_manual(values=c(color_scheme_get()$mid_highlight,
                             color_scheme_get()$light_highlight)) +
  theme_minimal() +
```

```
  theme(legend.position=c(.35, .85))+ labs(x="sigma")
grid.arrange(beta_plt, sigma_plt, ncol = 2)

min_m1 <- ppc_stat(y = data$y, yrep=
                     as_draws_matrix(
                       subset_draws(fit$draws(),
                                    regex =TRUE,
                                    variable = "y_pred")),
                   stat = "min")
max_m1 <- ppc_stat(y = data$y, yrep= as_draws_matrix(
  subset_draws(fit$draws(), regex =TRUE, variable = "y_pred")),
  stat = "max")
grid.arrange(min_m1, max_m1, nrow = 2)

ppc_stat(y = data$y, yrep= as_draws_matrix(
  subset_draws(fit$draws(), regex =TRUE, variable = "y_pred")),
  stat = "skewness")

ypreds_m1[sample(nrow(ypreds_m1), 50000), ] %>%
  ggplot(aes(time, y_pred, group = chain_iter)) +
  geom_line(alpha = 0.007) +
  geom_point(data = data,
             mapping = aes(t,y),
             inherit.aes = FALSE) +
  theme_minimal() +
  labs(x="t", y = "y/y_pred")

ppc_dens_overlay(y = data$y,
                 yrep = head(as_draws_matrix(
                   subset_draws(fit$draws(),
                                regex =TRUE,
                                variable = "y_pred")), 50))

plot(model1_loo)

# model 2 posterior

data_list2$only_prior = 0

fit2 <- mod2$sample(data_list2,seed = 365, refresh = 0)

mu_draws_post_m2 <- fit2$draws(c("mu[1]", "mu[10]", "mu[20]", "mu[30]", "mu[40]",
                                 "mu[50]", "mu[60]", "mu[70]", "mu[80]", "mu[90]",
                                 "mu[100]"))
pred_draws_post_m2 <- fit2$draws(c("y_pred[1]", "y_pred[10]",
                                   "y_pred[20]", "y_pred[30]",
                                   "y_pred[40]", "y_pred[50]", "y_pred[60]",
                                   "y_pred[70]", "y_pred[80]", "y_pred[90]",
                                   "y_pred[100]"))
model2_loo <- fit2$loo(save_psis=TRUE)

# class example code
ypreds_m2 <- fit2$draws() %>% reshape2::melt() %>%
```

```
  filter(str_detect(variable, "y_pred") ) %>%
  extract(col = variable, into = "ind",
          regex = "y_pred\\[([0-9]*)\\]",
          convert = TRUE)
ypreds_m2 <- ypreds_m2 %>%
  mutate(time = data_list$t[ind],
         chain_iter = glue::glue("chain {chain}, iteration {iteration}"),
         .keep = "unused") %>%
  rename(y_pred = value)

mu_posterior_m2<-melt(as_draws_matrix(
  subset_draws(fit2$draws(), regex =TRUE, variable = "mu"))) %>%
  mutate(variable = str_replace_all(variable,
                                    pattern="mu.*",
                                    replacement = "posterior"))
mu_prior_m2 <- melt(as_draws_matrix(
  subset_draws(prior_fit2$draws(), regex =TRUE, variable = "mu"))) %>%
  mutate(variable = str_replace_all(variable,
                                    pattern="mu.*",
                                    replacement = "prior"))

mu_comparison_df_m2 <- rbind(mu_posterior_m2, mu_prior_m2)

mu_plt_m2 <- ggplot(mu_comparison_df_m2,aes(x=value, fill = variable)) +
  geom_histogram(alpha=1) +
  scale_fill_manual(values=c(
    color_scheme_get()$mid_highlight,
    color_scheme_get()$light_highlight)) +
  theme_minimal() +  theme(legend.position=c(.25, .65)) + labs(x="mu")



alpha_posterior_m2<-melt(as_draws_matrix(
  subset_draws(fit2$draws(), regex =TRUE, variable = "alpha"))) %>%
  mutate(variable = str_replace_all(
    variable, pattern="alpha*", replacement = "posterior"))
alpha_prior_m2 <- melt(as_draws_matrix(
  subset_draws(prior_fit2$draws(), regex =TRUE, variable = "alpha"))) %>%
  mutate(variable = str_replace_all(
    variable, pattern="alpha*", replacement = "prior"))

alpha_comparison_df_m2 <- rbind(alpha_posterior_m2, alpha_prior_m2)

alpha_plt_m2 <- ggplot(alpha_comparison_df_m2,
                       aes(x=value, fill = variable)) +
  geom_histogram(alpha=1) +
  scale_fill_manual(values=c(
    color_scheme_get()$mid_highlight,
    color_scheme_get()$light_highlight)) +
  theme_minimal() +  theme(legend.position=c(.25, .65)) + labs(x="alpha")


beta_posterior_m2<-melt(as_draws_matrix(
```

```
  subset_draws(fit2$draws(), regex =TRUE, variable = "beta"))) %>%
  mutate(variable = str_replace_all(
    variable, pattern="beta*", replacement = "posterior"))
beta_prior_m2 <- melt(as_draws_matrix(
  subset_draws(prior_fit2$draws(), regex =TRUE, variable = "beta"))) %>%
  mutate(variable = str_replace_all(
    variable, pattern="beta*", replacement = "prior"))

beta_comparison_df_m2 <- rbind(beta_posterior_m2, beta_prior_m2)

beta_plt_m2 <- ggplot(beta_comparison_df_m2,aes(x=value, fill = variable))+
  geom_histogram(alpha=1) +
  scale_fill_manual(values=c(color_scheme_get()$mid_highlight,
                             color_scheme_get()$light_highlight)) +
  theme_minimal() +
  theme(legend.position=c(.25, .85)) +
  labs(x="beta")

grid.arrange(mu_plt_m2, alpha_plt_m2, ncol =2 )

sigma_posterior_m2<-melt(as_draws_matrix(
  subset_draws(fit2$draws(), regex =TRUE, variable = "sigma"))) %>%
  mutate(variable =
           str_replace_all(variable,
                           pattern="sigma*",
                           replacement = "posterior"))
sigma_prior_m2 <- melt(as_draws_matrix(
  subset_draws(prior_fit2$draws(),
               regex =TRUE, variable = "sigma"))) %>%
  mutate(variable =
           str_replace_all(variable,
                           pattern="sigma*",
                           replacement = "prior"))

sigma_comparison_df_m2 <- rbind(sigma_posterior_m2, sigma_prior_m2)

sigma_plt <- ggplot(sigma_comparison_df_m2,aes(x=value, fill = variable))+
  geom_histogram(alpha=1) +
  scale_fill_manual(values=c(
    color_scheme_get()$mid_highlight,
    color_scheme_get()$light_highlight)) +
  theme_minimal() +
  theme(legend.position=c(.35, .70))+
  labs(x="sigma")
grid.arrange(beta_plt, sigma_plt, ncol = 2)

min_m2 <- ppc_stat(y = data$y, yrep= as_draws_matrix(
  subset_draws(fit2$draws(), regex =TRUE, variable = "y_pred")), stat = "min")
max_m2 <- ppc_stat(y = data$y, yrep= as_draws_matrix(
  subset_draws(fit2$draws(), regex =TRUE, variable = "y_pred")), stat = "max")
grid.arrange(min_m2, max_m2, ncol = 2)
```

```
ppc_stat(y = data$y, yrep= as_draws_matrix(
  subset_draws(fit2$draws(),
               regex =TRUE, variable = "y_pred")),
  stat = "skewness")

ypreds_m2[sample(nrow(ypreds_m2), 50000), ] %>%
  ggplot(aes(time, y_pred, group = chain_iter)) +
  geom_line(alpha = 0.007) +
  geom_point(data = data,
             mapping = aes(t,y),
             inherit.aes = FALSE) +
  theme_minimal() +
  labs(x="t", y = "y/y_pred")

ppc_dens_overlay(y = data$y,
                 yrep = head(as_draws_matrix(
                   subset_draws(fit2$draws(),
                                regex =TRUE,
                                variable = "y_pred")), 100))

plot(model2_loo)

knitr::kable(loo_compare(model1_loo, model2_loo),
             caption = "loo_compare results")

# bonus plots

mu_1_draws <- prior_fit$draws(c("mu[1]"))
mu_1_draws_m1 <- mcmc_hist(mu_1_draws)

ypred_1_draws <- prior_fit$draws(c("y_pred[1]"))
ypred_1_draws_m1 <-mcmc_hist(ypred_1_draws)
grid.arrange(mu_1_draws_m1, ypred_1_draws_m1, ncol=2)


mu_1_draws <- fit$draws(c("mu[100]"))
mu_1_draws_m1 <- mcmc_hist(mu_1_draws)

ypred_1_draws <- fit$draws(c("y_pred[100]"))
ypred_1_draws_m1 <-mcmc_hist(ypred_1_draws)
grid.arrange(mu_1_draws_m1, ypred_1_draws_m1, ncol=2)

mcmc_hist(pred_draws_post)

log_plt <- data %>% ggplot(aes(x=t, y=log(y))) +
  geom_point()+
  labs(x="t",
       y="log(y)",
       title = "Scatterplot of log(y) vs t") +
  theme_minimal()

log_plt
```

```
mu_1_draws_m2 <- prior_fit2$draws(c("mu[1]"))
mu_1_draws_m2 <- mcmc_hist(mu_1_draws_m2)

ypred_1_draws_m2 <- prior_fit2$draws(c("y_pred[1]"))
ypred_1_draws_m2 <-mcmc_hist(ypred_1_draws_m2)
grid.arrange(mu_1_draws_m2, ypred_1_draws_m2, ncol=2)

mu_1_draws_m2 <- fit2$draws(c("mu[1]"))
mu_1_draws_m2 <- mcmc_hist(mu_1_draws_m2)

ypred_1_draws_m2 <- fit2$draws(c("y_pred[1]"))
ypred_1_draws_m2 <-mcmc_hist(ypred_1_draws_m2)
grid.arrange(mu_1_draws_m2, ypred_1_draws_m2, ncol=2)

mcmc_hist(pred_draws_post_m2)
```

Model 1 Stan code

```
data {
  int<lower=0> N;
  vector[N] y;
  vector[N] t;

  // prior params
  real mu_alpha;
  real mu_beta;
  real mu_sigma;
  real<lower = 0> tau_alpha;
  real<lower = 0> tau_beta;
  real<lower = 0> tau_sigma;

  int<lower=0, upper = 1> only_prior;
}

parameters {
  real alpha;
  real beta;
  real<lower=0> sigma;
}

transformed parameters{
  vector[N] mu = exp(alpha+beta*t);
}

model {
  // priors
  alpha ~ normal(mu_alpha, tau_alpha);
  beta  ~ normal(mu_beta, tau_beta);
  sigma ~ normal(mu_sigma, tau_sigma);

  // likelihood
  if (only_prior == 0){
    y ~ normal(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);
    y_pred[i] = normal_rng(mu[i], sigma);
  }
}
```

Model 2 Stan code

```
data {
  int<lower=0> N;
  vector[N] y; // log values of y
  vector[N] t;

  // prior params
  real mu_alpha;
  real mu_beta;
  real mu_sigma;
  real tau_alpha;
  real tau_beta;
  real tau_sigma;

  int<lower=0, upper = 1> only_prior;
}

parameters {
  real alpha;
  real beta;
  real<lower=0> sigma;
}

transformed parameters{
  vector[N] mu = alpha+beta*t;
}

model {
  // priors
  alpha ~ normal(mu_alpha, tau_alpha);
  beta  ~ normal(mu_beta, tau_beta);
  sigma ~ normal(mu_sigma, tau_sigma);

  // likelihood
  if (only_prior == 0){
      y ~ normal(mu, sigma);
  }
}
```

```
generated quantities {
  vector[N] log_lik;
  vector[N] y_pred;
  vector[N] y_lg;
  for (i in 1:N) {
    // change the scale of the support of log(y) distn
    log_lik[i] = lognormal_lpdf(exp(y[i]) | mu[i], sigma);
    // note that draws from normal distn are log predictions
    // so we will need to undo the log
    y_lg[i] = normal_rng(mu[i], sigma);
    y_pred[i] = exp(y_lg[i]);
  }
}
```