Bayesian Data Analysis - Assignment 1 (S1889112)

1a: UK Experts; $\mu \approx 1, \mu \in [0, 2]$. Conjugate prior. (5)

We know we have a normal likelihood with known variance, hence a conjugate prior is the normal. Since the mean is believed to be at 1, the hyperparameter for the mean is $\mu_0 = 1$. The experts believe the interval of possible values is [0, 2] so we design the prior such that there is roughly a 95% prior probability that μ is in these bounds. As such, we calibrate σ_0 by $1 + 2\sigma_0 = 2 \iff \sigma_0 = \frac{1}{2} \iff \sigma_0^2 = \frac{1}{4} \iff \tau_0 = 4$. Where $\tau_0 = \sigma^{-2}$

$$L(\mu|y, \sigma^2) \propto exp\left(\frac{-1}{2\sigma^2}\sum_{i=1}^{10}(y_i - \mu)^2\right), \quad p(\mu) \propto exp\left(\frac{-1}{2\sigma_0^2}(\mu - \mu_0)^2\right) \tag{1}$$

$$p(\mu|y) \propto exp\left(\frac{-1}{\frac{2\sigma^2\sigma_0^2}{\sigma^2+n\sigma_0^2}}\left(\mu^2 - 2\mu\frac{n\bar{y}\sigma_0^2 + \mu_0\sigma^2}{\sigma^2 + n\sigma_0^2}\right)\right) \propto exp\left(\frac{-1}{\frac{2\sigma^2\sigma_0^2}{\sigma^2+n\sigma_0^2}}\left(\mu - \frac{n\bar{y}\sigma_0^2 + \mu_0\sigma^2}{\sigma^2 + n\sigma_0^2}\right)^2\right) \tag{2}$$

This is the general (proportional) posterior for a normal-normal (known variance) model. It is the kernel of a normal with variance $\frac{\sigma^2\sigma_0^2}{\sigma^2+n\sigma_0^2}$ and mean $\frac{n\bar{y}\sigma_0^2+\mu_0\sigma^2}{\sigma^2+n\sigma_0^2}$. Hence we have the following posterior

$$\mu|y \sim N\left(\frac{n\bar{y}\sigma_0^2 + \sigma^2\mu_0}{\sigma^2 + n\sigma_0^2}, \frac{\sigma^2\sigma_0^2}{\sigma^2 + n\sigma_0^2}\right) = N\left(\frac{\tau_0\mu_0 + n\tau\bar{y}}{\tau_0 + n\tau}, \frac{1}{\tau_0 + n\tau}\right) = N(1.047, 0.231) \tag{3}$$

Calculations are in Appendix A.1

1b: US Experts, mixture prior (5)

The length of the interval that the experts suspect μ is suspected to be in is now 4, so we recalculate the variance with the same method as before. The US distribution is centered at 5. That is we have $5 + 2\sigma_1 = 7$, so we have $\sigma_1 = \frac{2}{2} \iff \sigma_1^2 = 1 \iff \tau_1 = 1$. So we have $(\mu_1, \sigma_1^2) = (5, 1)$.
For the weights below, I set $\pi_0 = \frac{2}{3}$, by using the proportion of the experts who support the initial prior (the UK prior). The mixture prior is as follows:

$$p(\mu) = \pi_0\frac{1}{\sqrt{2\pi\sigma_0^2}}exp\left(\frac{-1}{2\sigma_0^2}(\mu - \mu_0)^2\right) + (1 - \pi_0)\frac{1}{\sqrt{2\pi\sigma_1^2}}exp\left(\frac{-1}{2\sigma_1^2}(\mu - \mu_1)^2\right) \tag{4}$$

By conjugacy, the posterior can be written as a weighted sum of the individual posterior components.

$$\begin{aligned} p(\mu|y) &= wN\left(\frac{n\bar{y}\sigma_0^2 + \sigma^2\mu_0}{\sigma^2 + n\sigma_0^2}, \frac{\sigma^2\sigma_0^2}{\sigma^2 + n\sigma_0^2}\right) + (1 - w)N\left(\frac{n\bar{y}\sigma_1^2 + \sigma^2\mu_1}{\sigma^2 + n\sigma_1^2}, \frac{\sigma^2\sigma_1^2}{\sigma^2 + n\sigma_1^2}\right) && (5) \\ &= wN(1.047, 0.231) + (1 - w)N(4.153, 0.750) && (6) \\ &\approx 0.8967N(1.047, 0.231) + (1 - 0.8967)N(4.153, 0.750) && (7) \end{aligned}$$

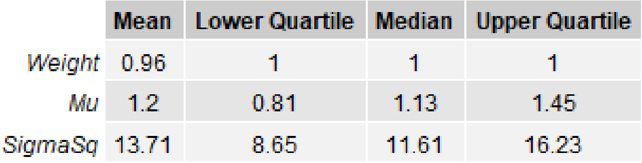
The weights were estimated by JAGS, with the exact expression as follows:

$$w = \frac{\pi_0 \int L(\mu|y, \sigma^2)p_1(\mu)d\mu}{\int[\pi_0 p_1(\mu) + (1 - \pi_0)p_2(\mu)]L(\mu|y, \sigma^2)d\mu} \tag{8}$$

1c: Posterior sample from unknown variance model (8)

We are considering the variance completely unknown (that is, we pay no attention to the fact it was previously 'known' at 30). For conjugacy, we use a gamma prior on tau. For vagueness, we specify the hyperparameters such that they are close to 0 (using the rate parameter). I choose $\tau \sim \Gamma(0.01, 0.01)$.

The code to obtain samples from the posterior is included in A.3



	Mean	Lower Quartile	Median	Upper Quartile
Weight	0.96	1	1	1
Mu	1.2	0.81	1.13	1.45
SigmaSq	13.71	8.65	11.61	16.23

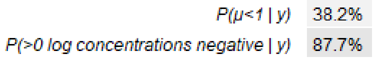
Figure 1: Summary Statistics for Posterior Samples

Figure 1 shows the results of sampling from the posterior. The 'Weight' row indicates the posterior weight on the UK expert prior is 96%. The means for μ and σ are both higher than the medians, suggesting a degree of positive skew. This is expected for σ^2 as it has a lower bound at 0. The reason μ is slightly skewed is because occasionally the posterior samples from the US posterior, which puts more density in the higher end of the distribution.

1d: Pr($\mu < 1$), Pr(At least one negative log-concentration) (6)

To find the probability μ is less than 1, we look at the proportion of posterior samples of μ that are less than 1.

To find the probability of at least one sample having a negative log concentration, we sample 5 times with the posterior parameters repeatedly, and use the fraction of samples which contain at least one negative log concentration. Note that this could also be done binomially, and the method for doing so is included in the appended code for this subsection. Results are in the table below:



P(μ<1 | y)	38.2%
P(>0 log concentrations negative | y)	87.7%

Figure 2: Posterior and Posterior Predictive Probabilities

1e: Weight as a prior, σ^2 known (6)

First note that the 'w' referred to in the question is the same as π_0 in the . I will use 'w' here for the prior to be consistent with the wording of the question. Note that p(w) = $I(w \in (0,1))$

The posterior is now:

$$\begin{aligned}
p(\mu, w|y) &= \frac{p(y|\mu,w)p(\mu|w)p(w)}{p(y)} = \frac{p(y|\mu,w)p(\mu|w)p(w)}{\int\int p(y|\mu,w)p(\mu|w)p(w)dwd\mu} &(9)\\
&= \frac{p(y|\mu,w)[wp_1(\mu)+(1-w)p_2(\mu)]p(w)}{\int\int p(y|\mu,w)p(\mu|w)p(w)dwd\mu} &(10)\\
&= \frac{wp(y|\mu,w)p(w)p_1(\mu)+(1-w)p(y|\mu,w)p(w)p_2(\mu)]}{\int\int p(y|\mu,w)p(\mu|w)p(w)dwd\mu} &(11)
\end{aligned}$$

Note that the individual components of the posterior can be written:

$$p_i(\mu,w|y)p_i(y) = p(y|\mu,w)p_i(\mu|w)p(w) = p(y|\mu,w)p_i(\mu)p(w) \tag{12}$$

Applying this to the numerator of the posterior yields:

$$\begin{aligned}
p(\mu,w|y) &= \frac{wp_1(\mu,w|y)\int\int p(y|\mu,w)p_1(\mu)p(w)dwd\mu + (1-w)p_2(\mu,w|y)\int\int p(y|\mu,w)p_2(\mu|w)p(w)dwd\mu}{\int\int p(y|\mu,w)p(\mu|w)p(w)dwd\mu} &(13)\\
&= w'p_1(\mu,w|y)+(1-w')p_2(\mu,w|y) &(14)\\
&= w'p_1(\mu|y)+(1-w')p_2(\mu|y) \quad \text{Since } p_1 \text{ does not depend on w} &(15)\\
&\text{Where} &(16)\\
w' &= w\frac{\int\int p(y|\mu,w)p_1(\mu)p(w)dwd\mu}{\int\int p(y|\mu,w)p(\mu|w)p(w)dwd\mu} \overset{\text{Terms independent of w}}{=} w\frac{\int p(y|\mu,w)p_1(\mu)\int p(w)dwd\mu}{\int\int p(y|\mu,w)p(\mu|w)p(w)dwd\mu} &(17)\\
&= w\frac{\int p(y|\mu,w)p_1(\mu)d\mu}{\int\int p(y|\mu,w)p(\mu|w)p(w)dwd\mu} &(18)
\end{aligned}$$

Which is the same as (8).

To get the marginals, we integrate over the other variable, so:

$$\begin{aligned}
p(\mu|y) &= \int p(\mu,w|y)dw = p_1(\mu|y)\int w'dw + p_2(\mu|y)\int(1-w')dw &(19)\\
&= p_1(\mu|y)\frac{\int p(y|\mu,w)p_1(\mu)d\mu}{\int\int p(y|\mu,w)p(\mu|w)p(w)dwd\mu}\int wdw + p_2(\mu|y)\frac{\int p(y|\mu,w)p_2(\mu)d\mu}{\int\int p(y|\mu,w)p(\mu|w)p(w)dwd\mu}\int(1-w)dw &(20)\\
&= 0.5\frac{\int p(y|\mu,w)p_1(\mu)d\mu}{\int\int p(y|\mu,w)p(\mu|w)p(w)dwd\mu}p_1(\mu|y) + 0.5\frac{\int p(y|\mu,w)p_2(\mu)d\mu}{\int\int p(y|\mu,w)p(\mu|w)p(w)dwd\mu}p_2(\mu|y) &(21)\\
p(w|y) &= \int p(\mu,w|y)d\mu = w'\int p_1(\mu|y)d\mu + (1-w')\int p_2(\mu|y)d\mu = w'+1-w' = 1. &(22)
\end{aligned}$$

The marginal distribution for w is unchanged. The marginal distribution for μ is a weighted sum of the individual conjugate posteriors.

Question 2

2a: Prior Choice (6)

The conjugate to the Poisson is the gamma distribution. It is said that the value of λ has estimated values ranging from 0.01 to 1000, suggesting we have very little information about the values λ can take. As such, I will use a vague gamma prior, meaning hyperparameter values near 0. In particular, $\lambda \sim \Gamma(0.001, 0.001)$.
The reason for using such low values, is that if we observe a small number of signals then (say if we used a=b=0.1) the hyperparameters may act as more informative than we wish them to, given the form of the posterior (as seen in equation (27). This comes back to the issue of the gamma having a peak at just above 0 as the parameters tend to 0. This may pull values of λ to the left if the prioris overly peaked.
The particular problem comes with using the rate parameter. Since runs are fractions of years, a value of 0.1 can take up a fairly large proportion of the posterior rate parameter. For example, for half a year, we use a conversion parameter of 0.5. The ratio of the prior to the posterior rate parameter is $0.1/0.6 = 1/6 \approx 17\%$. This is quite a large proportion of the posterior parameter considering we wish the prior to be as uninformative as possible.

Note that this reasoning above is not dependent on any data observed. This exact reasoning can be used before any run is started, as this is entirely based on knowing the length of various runs in advance.

2b: Posterior for λ (5)

We first note that λ is in 'per-year' units. As such, we add a constant κ to the likelihood to account for the varying time periods throughout. That is to say $y_i \sim Poi(\kappa_i \lambda)$, where κ_i converts the yearly λ to a level consistent with the time period measured over. For example, for O1, $\kappa = \frac{3}{12}$, as this converts the rate to units of "per 3 months". Proportionally, the likelihood, prior, and posterior are as follows (for a single observation):

$$f(y|\kappa, \lambda) \propto (\lambda\kappa)^y exp(-\lambda\kappa) \propto (\lambda)^y exp(-\lambda\kappa) \tag{23}$$

$$p(\lambda) \propto \lambda^{a-1} exp(-b\lambda) \tag{24}$$

$$p(\lambda|y) \propto \lambda^{a+y-1} exp(-\lambda(b+\kappa)) \tag{25}$$

$$\lambda|y \sim \Gamma(a+y, b+\kappa) \tag{26}$$

So for multiple observations, the posterior is:

$$\lambda|y \sim \Gamma\left(a + \sum_{i=1}^{n} y_i, b + \sum_{i=1}^{n} \kappa_i\right) \tag{27}$$

For O1, we have $y_1 = 1, \kappa_1 = \frac{3}{12}$ and the hyperparameters a = b = 0.001, so:

$$\lambda|y \sim \Gamma(0.001 + 3, 0.001 + 3/12) = \Gamma(3.001, 0.251) \tag{28}$$

Hence the mean, variance, and standard deviation can all be calculated exactly. The mean is 11.956 ($\frac{3.001}{0.251}$), the standard deviation is 5.030 ($\frac{3.001}{0.251^2}$), and the symmetric credible interval is [2.467, 28.790], which can be obtained using the `qgamma` function in R. The posterior is plotted in Figure 3 with the posterior mean, 95% symmetric credible interval, and the mean ± the posterior standard deviation.

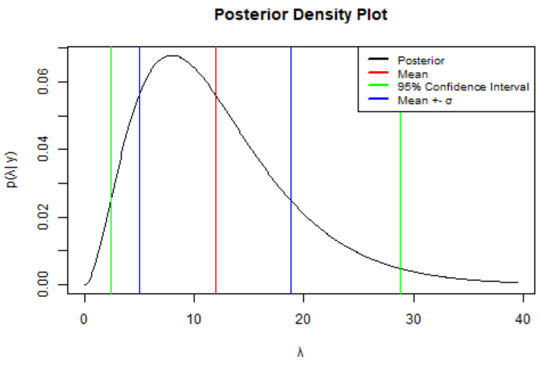


Figure 3: Posterior Density and Summary Statistics

2c: Pr($\lambda > 15$|y) (4)

The posterior probability that $\lambda > 15$ is 11.5%, which was found using the `pgamma` function in R. Code is in B.2.

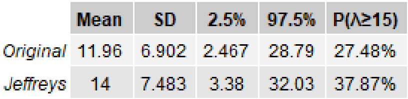
2d: Previous, using Jeffreys' Prior (5)

Jeffreys prior is defined as $p_J(\lambda) \propto \sqrt{I(\lambda|y)} = \sqrt{-E\left[\frac{d^2}{d\lambda^2}log(L(\lambda|y))\right]}$. Following the steps through, we get that for the Poisson, $p_J(\lambda) \propto \lambda^{-1/2} = \lambda^{1/2-1}exp(-0\lambda)$. This last equality shows that Jeffreys prior is an improper gamma prior, such that $p_J(\lambda) = \Gamma(\frac{1}{2}, 0)$. Note that the rate parameter being 0 nicely coincides with our reasoning in 2a about the influence of the rate parameter, completely removing the prior influence.

Due to the Jeffreys prior being a gamma, we can update it in the same way as in equation (27), hence under Jeffreys' prior we have the following posterior:

$$\lambda|y \sim \Gamma(0.5 + 3, 0 + 3/12) = \Gamma(3.5, 0.25) \tag{29}$$

The comparison is seen in Figure 4



	Mean	SD	2.5%	97.5%	P(λ≥15)
Original	11.96	6.902	2.467	28.79	27.48%
Jeffreys	14	7.483	3.38	32.03	37.87%

Figure 4: Posterior Summary (Jeffreys vs Original)

The values for the mean and standard deviation are both higher than previously, leading to a greater probability that $\lambda > 15$ due to the zero lower bound on λ (so there is a higher chance of extremely high values). The symmetric credible interval is wider than previously. Results for the Jeffreys posterior are in Figure 5, with the previous mean also included.

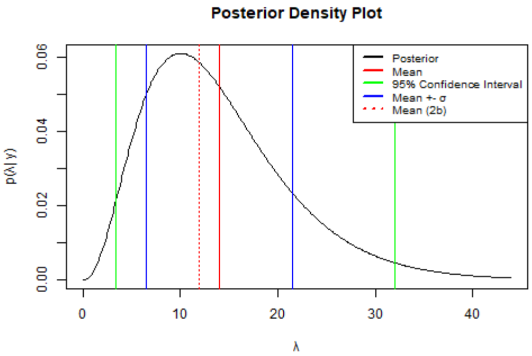


Figure 5: Posterior Density and Summary Statistics with Jeffreys Prior

2e: Posterior Predictive on O2 data (7)

Before defining the posterior predictive, let us define a' and b' as the hyperparameters of the posterior distribution from the O1 data. In predicting $\tilde{y}$ we have a new likelihood, as the conversion parameter κ changes. The posterior predictive is then as follows ($\tilde{\kappa}$ is the conversion parameter of the value we attempt to predict).

$$p(\tilde{y}|y) = \int f(\tilde{y}|\lambda)p(\lambda|y)d\lambda = \int \frac{(\lambda\tilde{\kappa})^{\tilde{y}}exp(-\lambda\tilde{\kappa})}{\tilde{y}!}\frac{(b')^{a'}}{\Gamma(a')}\lambda^{a'-1}exp(-b'(\lambda))d\lambda \tag{30}$$

$$= \frac{\tilde{\kappa}^{\tilde{y}}}{\tilde{y}}\frac{(b')^{a'}}{\Gamma(a')}\int \lambda^{a'+\tilde{y}-1}exp(-\lambda(\tilde{\kappa}+b'))d\lambda \tag{31}$$

$$= \frac{(a'+\tilde{y}-1)!}{\tilde{y}!(a'-1)!}\left(\frac{\tilde{\kappa}}{\tilde{\kappa}+b'}\right)^{\tilde{y}}\left(\frac{b'}{\tilde{\kappa}+b'}\right)^{a'} \tag{32}$$

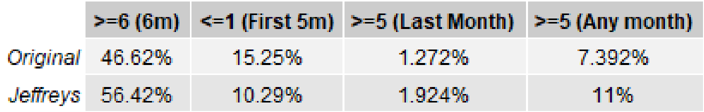
Note that R defines the negative binomial (where size=r is the number of successful trials, and p is the probability of success in each trial) density as:

$$P(X = x) = \frac{\Gamma(x+r)}{\Gamma(r)x!}p^n(1-p)^x = \frac{(r+x-1)!}{x!(r-1)!}(1-p)^x p^r \tag{33}$$

We compare equations (32) and (33), and notice the terms match up exactly. So we have that r = a', x = $\tilde{y}$, and $p = \frac{b'}{\tilde{\kappa}+b'}$. Hence using the parametrisation given by R, we have:

$$\tilde{y}|y \sim NB\left(r = a', p = \frac{b'}{\tilde{\kappa}+b'}\right) \tag{34}$$

As such we can calculate all the predictive probabilities exactly, using `pnbinom`. The code is in the appendix, and results are in Figure 6, rounded to 4 sf. To calculate the probability that there are more than 5 signals in any individual month, I considered p = Pr(At least 5 signals in any given month) the same as Pr(At least 5 signals in the last month) by independence. Then the probability of observing 4 or less signals in one month is 1-p. Lastly, the probability of observing more than 5 signals in any of the 6 months is 1 - Pr(Observing less than 5 signals in all 6 months), which is $1 - (1 - p)^6$.



	>=6 (6m)	<=1 (First 5m)	>=5 (Last Month)	>=5 (Any month)
Original	46.62%	15.25%	1.272%	7.392%
Jeffreys	56.42%	10.29%	1.924%	11%

Figure 6: Impact of Prior on Posterior Predictive

The results are moderately impacted by the choice of prior. The Jeffreys prior assigns more weight to the higher end of the distribution, resulting in higher probabilities for columns 1, and 3 (I do not consider 4 as extra evidence, since it is entirely based on column 3).

I agree with the experts, given the probabilities in Figure 6. Considering column 4 in particular, observing at least 5 signals in a single month at any point in a 6-month period is only 11% likely (at most, given both priors). Given that we actually observed this, combined with observing less than 1 signal in 5 months suggests that although the average over 6 months is quite stable, within each month the rate parameter has some variability.

2f: O3 (8)

I will use the original prior, as it has less impact on the hyperparameters of the posterior. To update the prior with O2 information, we use equation (27). As a result, the posterior is now:

$$\lambda|y \sim \Gamma(3.001 + 6, 0.251 + 6/12) = \Gamma(9.001, 0.751) \tag{35}$$

The predictive distributions for both n_i are identical, as they are over the same length of time (hence have the same conversion parameter. So the predictive distributions are as follows:

$$n_i \sim NB\left(r = 9.001, p = \frac{0.751}{0.5 + 0.751}\right) \tag{36}$$

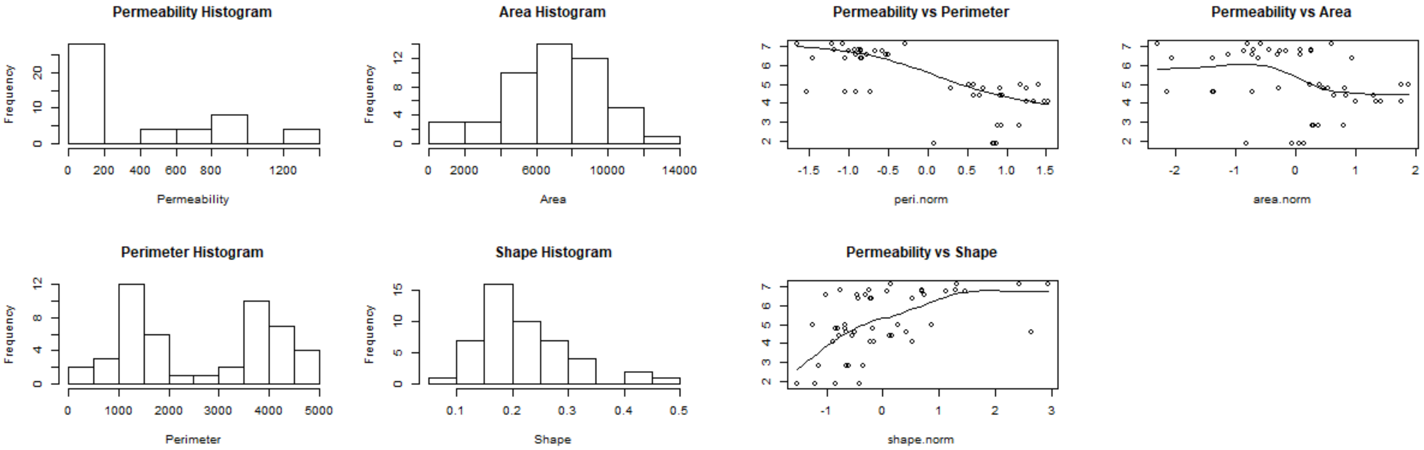
To conclude time heterogeneity, I will look for a difference 'D', such that the probability the difference is greater than 'D' is less than 5%. To find this value, I simulate n_1 and n_2 from the posterior predictive distributions, take the absolute difference, sort the values, and look at the value at the 95th percentile. This yields a value of 9. So if we observe a value of at least 9, we can be reasonably sure that the rate is not constant over time.

An alternative way of hypothesis testing is using Bayes factors. We could create 2 models (one with homogeneous time effects, and one with heterogeneous time effects), then examine the Bayes factor (Ratio of posterior odds to prior odds) between the models, and see if either conclusion is supported, using the rule of thumb that $BF_{ij} > 3$ means there is positive evidence for model i over model j. This could be done by having one model where λ is constant, and another where its hyperparameters depend on time.

Question 3: Bayesian Linear Regression

Exploratory Data Analysis

Firstly, I begin by looking at the distribution of each variable via histograms (as all variables are continuous). This is seen in Figure 7 (a). We see that the permeability is right-skewed, with some very high values. Area appears approximately normal, perimeter is bimodal, and the shape is also right-skewed. Since the covariates are on vastly different scales, we standardise them, which also helps with convergence. We also use log permeability to reduce the impact of the extreme values of the permeability. Additionally, since linear regression can output any value on the real line, we ensure that predictions for permeability are in $[0, \infty]$ with the logarithm.



a: Univariate Analysis

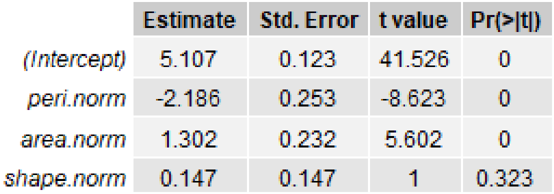
b: Bivariate Analysis

Figure 7: Histograms and Scatterplots

Next, I look at scatterplots of each variable against our dependent variable as seen in Figure 7 (b). There is seemingly a linear negative relationship between the perimeter and permeability. It appears as if after a standardised area of 0, the log permeability shifts downwards indicating a dummy effect. There appears to be a diminishing association between the shape and the permeability, but this is likely due to the outlier dragging the curve down. Given the limited number of observations (48), I will only use 3 covariates: The standardised perimeter, the standardised area being greater than 0, and the standardised shape. The reason for not using a dummy for the standardised area is that bivariate relationships ignore the effect of exogenous variables (i.e. the other covariates). Once these relationships are accounted for, it could be that the dummy effect disappears.

Figure 8 displays the results of a classical linear regression with the selected variables, which will be compared to the Bayesian regression later. It appears that standardised perimeter and area have a statistically significant association with the log permeability, while the shape has no effect.

Figure 9 shows the residual plots of the same regression. Using the curve as a guide, there appears to be a quadratic relationship between the residuals and the fitted values. On closer inspection there appear to be 3 clusters of points each with their own negative relationship. Additionally, the QQ plot suggests a departure from normality, particularly when considering the quantiles outside [-1, 1].



	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	5.107	0.123	41.526	0
peri.norm	-2.186	0.253	-8.623	0
area.norm	1.302	0.232	5.602	0
shape.norm	0.147	0.147	1	0.323

Figure 8: Summary of Classical Linear Regression

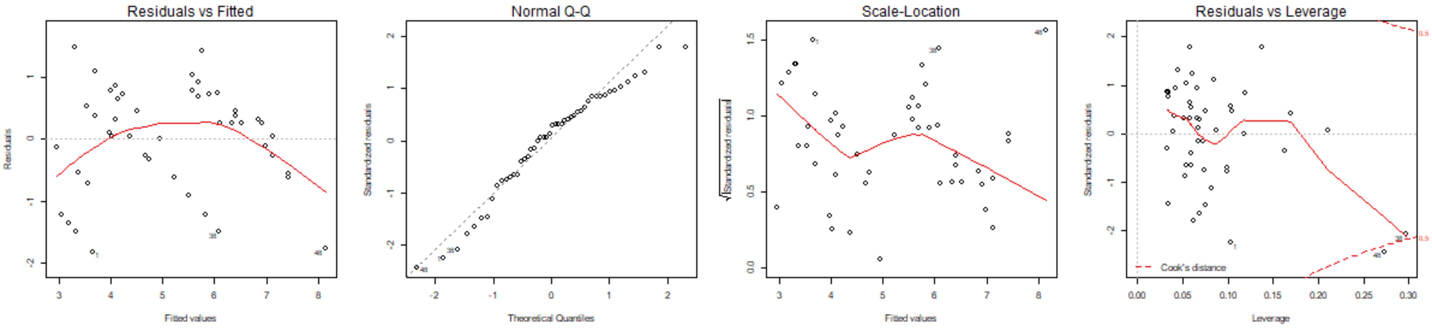


Figure 9: Residual Plots for Classical Linear Regression

Initial Model Specification, Fit, and Convergence

Given the departure from normality indicated by the QQ plot, and the outliers, I will use a t-distribution for the likelihood with specified mean (μ_i) and precision (τ). That is:

$$y_i|\mu_i, \tau, \nu, x_i \overset{ind}{\sim} t(\nu, \mu_i, \tau) \tag{37}$$

$$\mu_i = \beta_1 + \beta_2 Peri_i + \beta_3 Area_i + \beta_4 Shape_i \tag{38}$$

$$L(y|\mu_i, \tau, \nu) = \prod_{i=1}^{48} p(y_i|\mu_i, \tau, \nu, x_i) \tag{39}$$

I will use priors on β_i (normal) and τ (gamma) to be vague. For now, I consider $\nu = 5$.

$$\beta_j \sim N(0, 100000) \quad j = 1, 2, 3, 4 \tag{40}$$

$$\tau \sim \Gamma(0.1, 0.1) \tag{41}$$

$$\nu = 5 \tag{42}$$

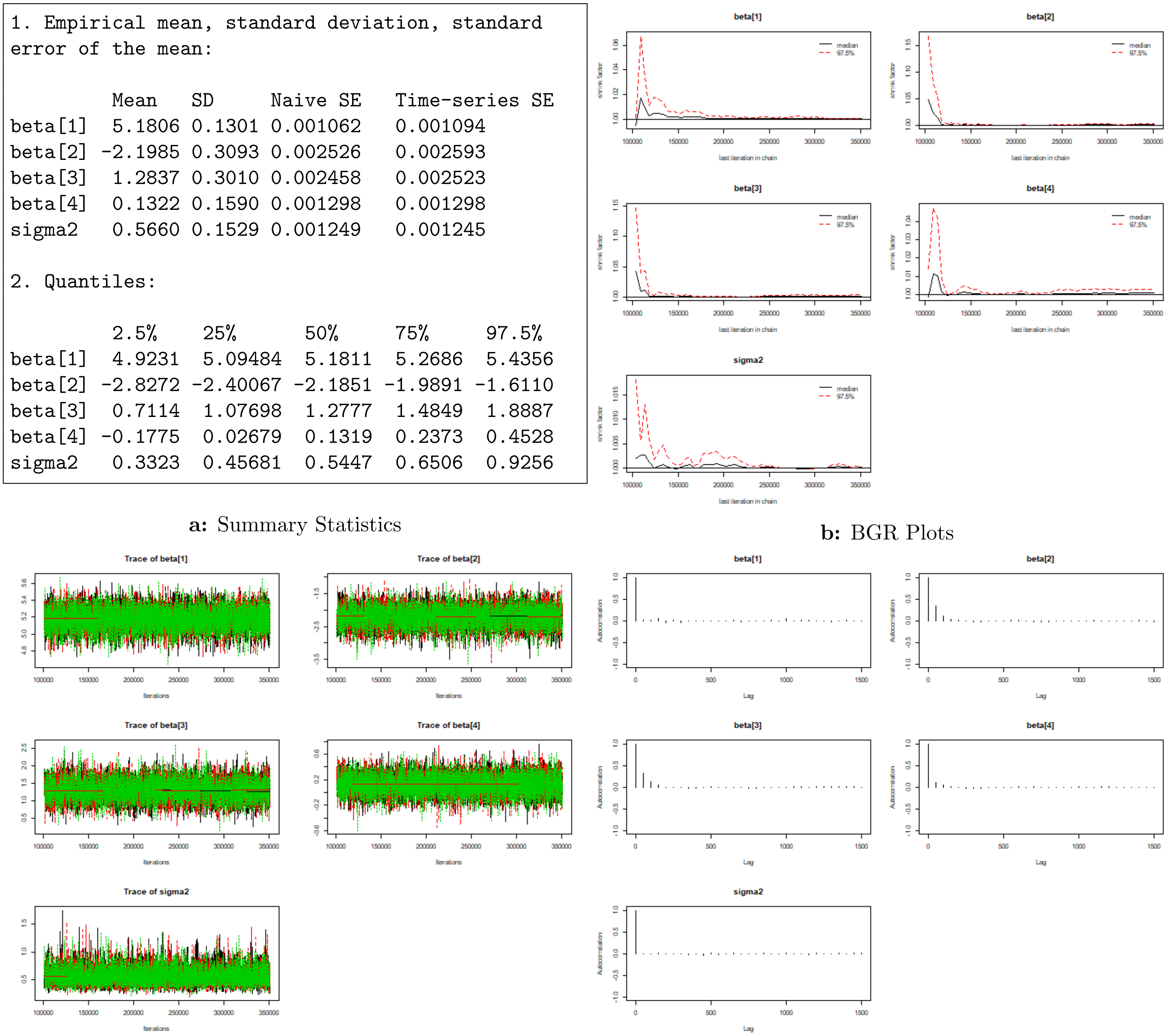
To sample from the posterior, I used a burn-in of 100000, and 250000 iterations, with a thinning interval of 50. This left 5000 observations. 3 chains were used, with initialisation left to JAGS.

Results for summary statistics for this model, and convergence diagnostics are available in figure 10.

We see that the 95% credible interval for β_2, and β_3 does not contain 0, suggesting the respective associations are non-zero. This indicates that the association of log permeability with the perimeter and area is negative and positive respectively. On the other hand, the association with the shape is potentially irrelevant controlling for the other variables, since its credible interval contains 0.

These results are very similar to the classical model, the direct comparison shown in Figure 11. If we consider the framework of classical hypothesis tests (i.e. if 0 is not in the 95% interval, there is some effect at the 5% level), then both models reject the null hypothesis of no effect for the same variables - namely the standardised perimeter and area. Additionally, the coefficients all have reasonably similar values. Lastly, the standard deviations of the Bayesian estimates are larger than (but similar to) the standard errors of the classical model, reflecting our uncertainty about the estimates.

With regards to convergence, the BGR statistic is near 1 for all parameters, which indicates convergence. The trace plots also indicate convergence (the lines all move around the same region, covering the space well). Additionally, we have reasonably low autocorrelation for all parameters, which is reflected in the effective sample sizes (for one chain) of 5000, 2867, 2897, 4018, and 5000 (not shown) It is clear there was some degree of autocorrelation for β_2, β_3 and β_4 but this is low as indicated by the plots.



```
1. Empirical mean, standard deviation, standard
error of the mean:

          Mean   SD     Naive SE   Time-series SE
beta[1]   5.1806 0.1301 0.001062   0.001094
beta[2]  -2.1985 0.3093 0.002526   0.002593
beta[3]   1.2837 0.3010 0.002458   0.002523
beta[4]   0.1322 0.1590 0.001298   0.001298
sigma2    0.5660 0.1529 0.001249   0.001245

2. Quantiles:

          2.5%    25%     50%     75%     97.5%
beta[1]   4.9231  5.09484  5.1811  5.2686  5.4356
beta[2]  -2.8272 -2.40067 -2.1851 -1.9891 -1.6110
beta[3]   0.7114  1.07698  1.2777  1.4849  1.8887
beta[4]  -0.1775  0.02679  0.1319  0.2373  0.4528
sigma2    0.3323  0.45681  0.5447  0.6506  0.9256
```

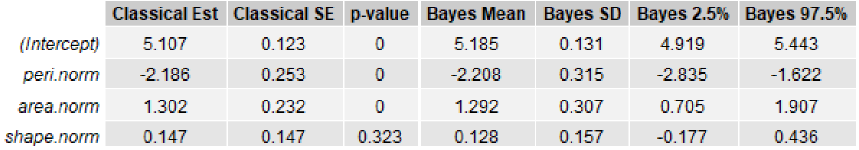
a: Summary Statistics

b: BGR Plots

c: Trace Plots

d: Autocorrelation Plots

Figure 10: Model 1 ($\nu = 5$) Summary and Convergence



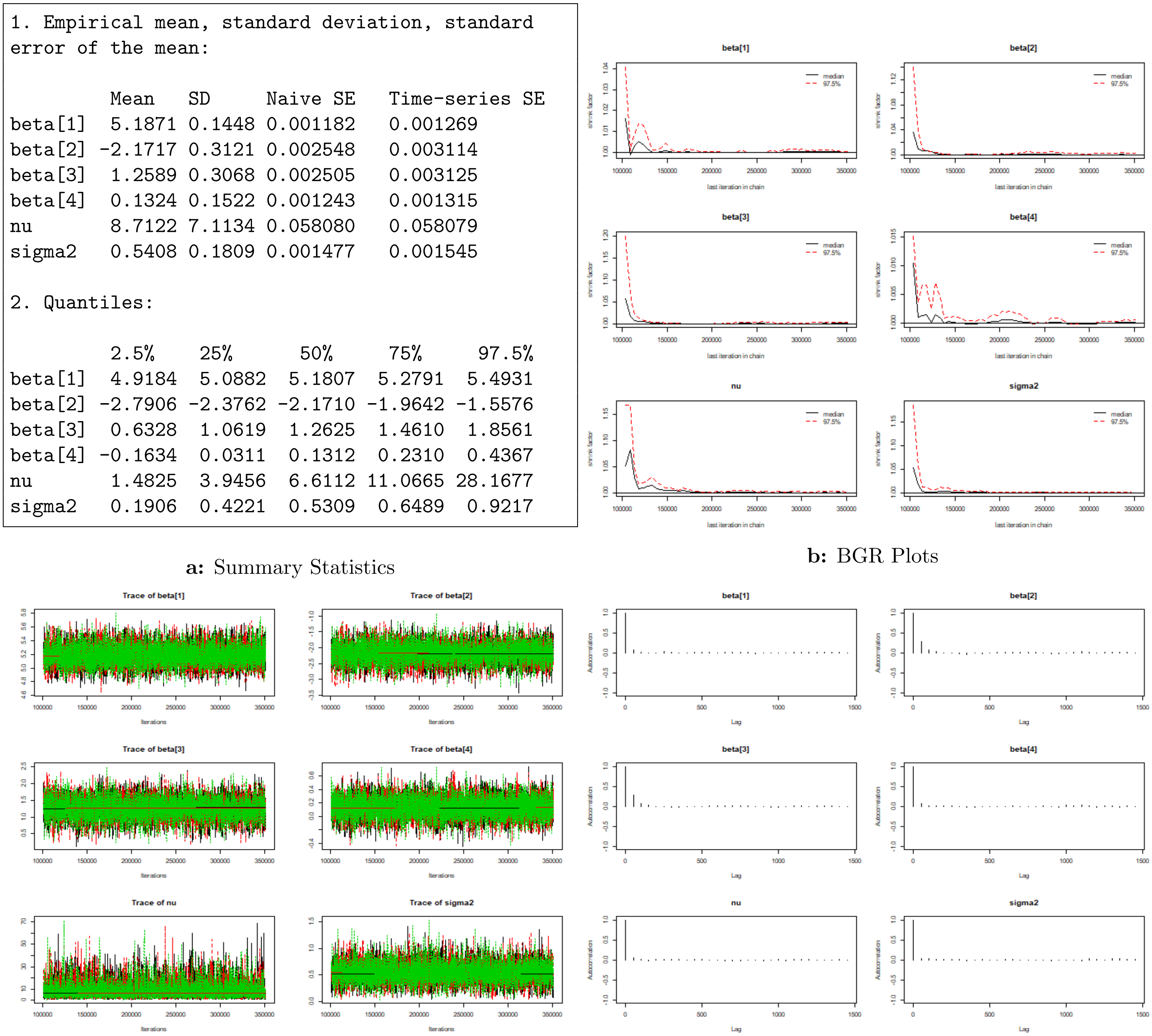
	Classical Est	Classical SE	p-value	Bayes Mean	Bayes SD	Bayes 2.5%	Bayes 97.5%
(Intercept)	5.107	0.123	0	5.185	0.131	4.919	5.443
peri.norm	-2.186	0.253	0	-2.208	0.315	-2.835	-1.622
area.norm	1.302	0.232	0	1.292	0.307	0.705	1.907
shape.norm	0.147	0.147	0.323	0.128	0.157	-0.177	0.436

Figure 11: Bayesian-Classical Comparison

Prior Sensitivity

Following on from this, we check how sensitive the model is to different priors. The errors remain t-distributed. The degrees of freedom ν are allowed to vary with a vague prior ($\nu \sim \Gamma(0.1, 0.1)$. A vague, flat prior is used for the precision ($\tau \sim U(0, 100000)$, and the β_j remain normally distributed, but with a lower variance such that $\beta_j \sim N(0, 1000)$.

I maintain the same burn-in and sample size. Results are in Figure 12, with a side-by-side comparison with the previous prior in Figure 13



```
1. Empirical mean, standard deviation, standard
error of the mean:

         Mean    SD     Naive SE   Time-series SE
beta[1]  5.1871 0.1448 0.001182   0.001269
beta[2] -2.1717 0.3121 0.002548   0.003114
beta[3]  1.2589 0.3068 0.002505   0.003125
beta[4]  0.1324 0.1522 0.001243   0.001315
nu       8.7122 7.1134 0.058080   0.058079
sigma2   0.5408 0.1809 0.001477   0.001545

2. Quantiles:

         2.5%    25%     50%     75%     97.5%
beta[1]  4.9184  5.0882  5.1807  5.2791  5.4931
beta[2] -2.7906 -2.3762 -2.1710 -1.9642 -1.5576
beta[3]  0.6328  1.0619  1.2625  1.4610  1.8561
beta[4] -0.1634  0.0311  0.1312  0.2310  0.4367
nu       1.4825  3.9456  6.6112 11.0665 28.1677
sigma2   0.1906  0.4221  0.5309  0.6489  0.9217
```

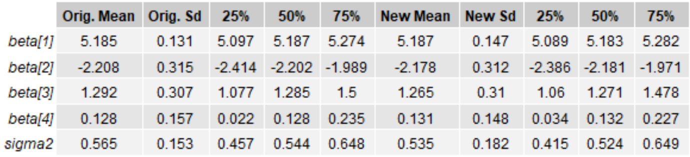
a: Summary Statistics

b: BGR Plots

c: Trace Plots

d: Autocorrelation Plots

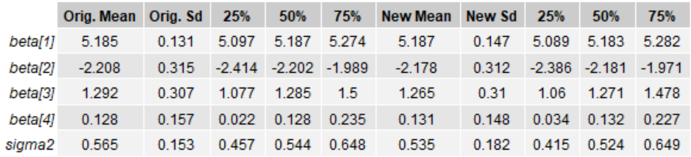
Figure 12: Model 2 ($\nu \sim \Gamma(0.1, 0.1)$; $\tau \sim U(0, 100000)$; $\beta_j \sim N(0, 1000)$) Summary and Convergence



	Orig. Mean	Orig. Sd	25%	50%	75%	New Mean	New Sd	25%	50%	75%
beta[1]	5.185	0.131	5.097	5.187	5.274	5.187	0.147	5.089	5.183	5.282
beta[2]	-2.208	0.315	-2.414	-2.202	-1.989	-2.178	0.312	-2.386	-2.181	-1.971
beta[3]	1.292	0.307	1.077	1.285	1.5	1.265	0.31	1.06	1.271	1.478
beta[4]	0.128	0.157	0.022	0.128	0.235	0.131	0.148	0.034	0.132	0.227
sigma2	0.565	0.153	0.457	0.544	0.648	0.535	0.182	0.415	0.524	0.649

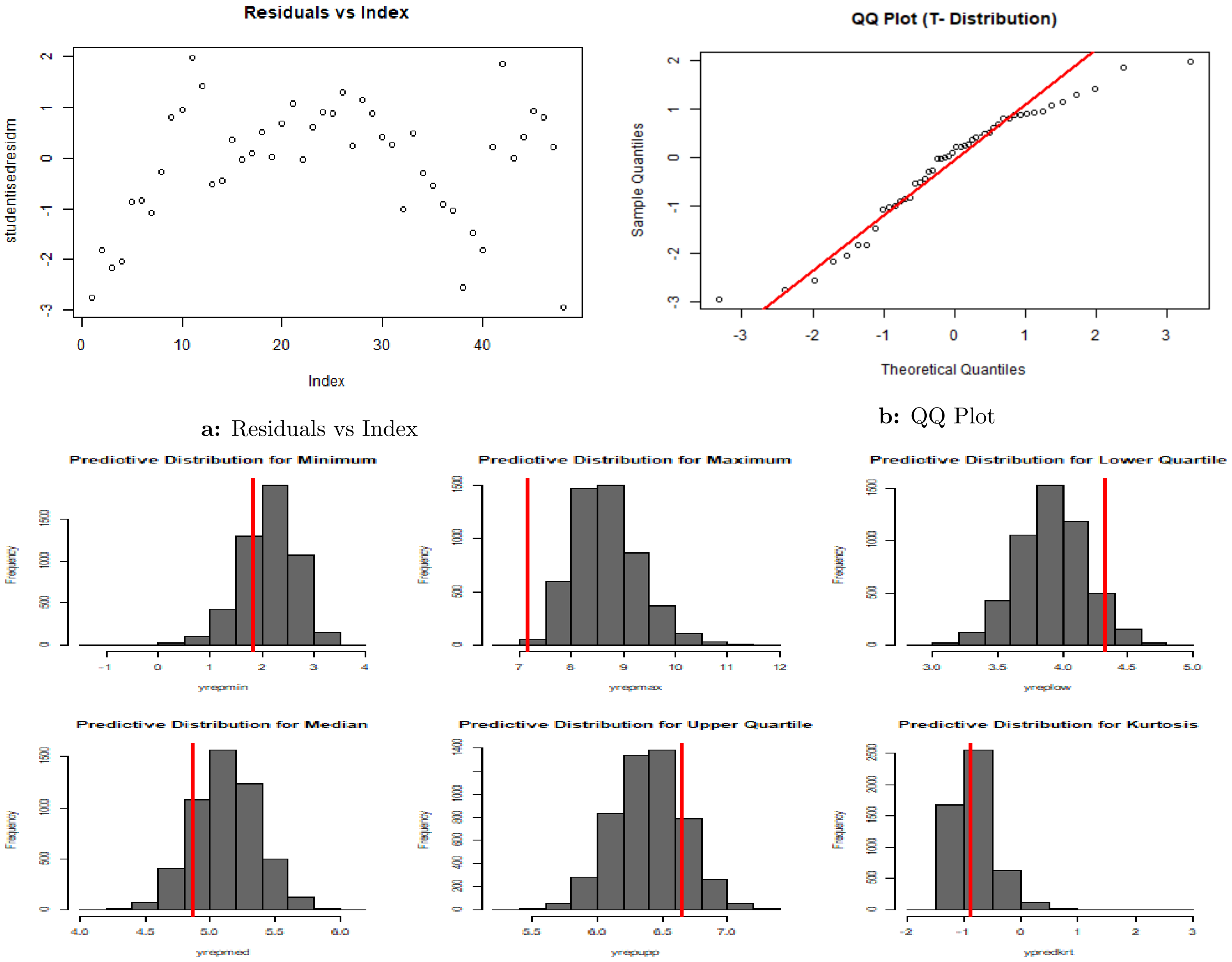
Figure 13: Prior Comparison (t-distributions: Model 1 vs Model 2)

The diagnostics give us no reason to suspect a lack of convergence. Comparing both priors, Figure 13 shows the new model changes very little. The means, standard deviations, and quartiles are almost identical. In general, the parameters for the new prior have slightly higher standard deviation (β_4 being the exception), but the difference is fairly small. This may be due to the similar distribution, however the results are similar with a normal likelihood (results in Figure 14) and the same priors as Model 2. Though the normal likelihood changes the results more than the t-distribution. There is a lower standard deviation on all parameters, but the overall estimated σ^2 is higher. Given the results of the prior checks, I conclude the model is not sensitive to the prior.



	Orig. Mean	Orig. Sd	25%	50%	75%	New Mean	New Sd	25%	50%	75%
beta[1]	5.185	0.131	5.097	5.187	5.274	5.187	0.147	5.089	5.183	5.282
beta[2]	-2.208	0.315	-2.414	-2.202	-1.989	-2.178	0.312	-2.386	-2.181	-1.971
beta[3]	1.292	0.307	1.077	1.285	1.5	1.265	0.31	1.06	1.271	1.478
beta[4]	0.128	0.157	0.022	0.128	0.235	0.131	0.148	0.034	0.132	0.227
sigma2	0.565	0.153	0.457	0.544	0.648	0.535	0.182	0.415	0.524	0.649

Figure 14: Comparison of Priors (t-distribution vs Normal)



a: Residuals vs Index

b: QQ Plot

c: Posterior Predictive Checks

Figure 15: Model 1 ($\nu = 5$) Model Checks

Unfortunately, the residuals seem to illustrate a roughly quadratic relationship with the index, indicating some degree of serial correlation. The QQ plot is adequate in [-2.5, 1], but there is a large outlier in the bottom left, and at the upper end the points diverge from the t(5). The posterior predictive checks are more promising, with the generated samples capturing the summary statistics for the most part. However the maximum value appears to be largely overestimated. Given the performance of the posterior predictive distribution, it seems there is some potential for this model - but given the broken assumption of serial correlation, and the poor QQ plot at the upper end, I conclude alternative models should be explored. Perhaps involving dummies and interaction terms once more observations are available to work with.

A Appendix: Question 1

A.1 1a

```r
#### Load

library(rjags)
library(gridExtra)

##### 1a (Initial) #####

y <- c(-0.566,  3.74, 5.55, -1.90, -3.54,
         5.16 , -1.76, 4.08,  4.62,  0.732)
n <- length(y)
mu0  <- 1
prec <- 1/30
prec0 <- 4

ybar <- mean(y)


norm.post.calc <- function(y, prec, prec.h, mu.h){
  # y: Data
  # prec:   Known precision
  # prec.h: Hyperparameter for precision of mu
  # mu.h:    Hyperparameter for mean of mu
  ybar <- mean(y)
  n    <- length(y)

  ## Mean:
  post.mu <- (prec.h*mu.h + n*prec*ybar)/(prec.h + n*prec)
  ## Variance:
  post.var <- 1/(prec.h + n*prec)
  ## Return
  c(post.mu, post.var)
}

norm.post.calc(y, prec = prec, prec.h =  prec0, mu.h = mu0)
```

A.2 1b

```r
### New hyperparameters:
prec1  <- 1
mu1    <- 5

### Posterior hyp. of US prior

norm.post.calc(y, prec=prec, prec.h = prec1, mu.h = mu1)


### Weight estimation in JAGS
mus  <- c(mu0, mu1)
taus <- c(prec0, prec1)
pis  <- c(2/3, 1/3)


mod.1b <- "model{
  # Likelihood
  for (i in 1:n){
    y[i] ~ dnorm(mu, tau)
  }

  # Prior
  mu0 ~ dnorm(mus[1], taus[1])
  mu1 ~ dnorm(mus[2], taus[2])
  w   ~ dbinom(pis[1], 1)
  mu = w*mu0 + (1-w)*mu1

}"

data <- list(y=y, n=n,
             mus = mus, taus = taus, pis = pis,
             tau = 1/30)
```

```

model.1b <- jags.model(textConnection(mod.1b), n.chains = 3, data = data)
update(model.1b, 10000)
samp.1b <- coda.samples(model.1b, variable.names = c('mu', 'w'), n.iter = 20000)
```

A.3 1c

```

## Hyperparameters on tau
a <- 0.01
b <- 0.01

## JAGS MODEL:

mod.1c <- "model{

## Likelihood

for (i in 1:n) {
  y[i] ~ dnorm(mu, tau)
}


## Prior on mu

mu1 ~ dnorm(mus[1], taus[1])
mu2 ~ dnorm(mus[2], taus[2])
I   ~ dbinom(pis[1], 1) # Choose distribution 1 with probability 1.
mu = I*mu1 + (1-I)*mu2

## Prior on tau
tau ~ dgamma(a, b)

## Convert precision to variance
sig2 = 1/tau

}"

# Data to feed to JAGS

data <- list(y = y, n = n,
             mus = mus, taus = taus, pis = pis,
             a   = a, b   = b)


# Compile and initialise model

model.1c <- jags.model(textConnection(mod.1c), n.chains = 5, data = data)

# Update: Select burn-in of 10000
update(model.1c, 10000)

# Get samples

res.1c <- coda.samples(model.1c,
                    variable.names = c("mu", "sig2", 'I'), n.iter = 10000)


##### 1c: Posterior mean, median, lower and upper quartiles #####

chain1 <- res.1c[[1]]
summ.1c <- as.data.frame(t(rbind(apply(chain1, 2, mean),
                   apply(chain1, 2, quantile))))
summ.1c[, c('0%', '100%')] <- NULL
colnames(summ.1c) <- c('Mean', 'Lower Quartile', 'Median', 'Upper Quartile')
rownames(summ.1c) <- c('Weight', 'Mu', 'SigmaSq')

#### Send to Image ###
png(filename = 'Q1/1cSummary.png', width=380,height=100)
grid.table(round(summ.1c, 2))
dev.off()
```

A.4 1d

```
## JAGS MODEL:

mod.1d <- "model{

## Likelihood

for (i in 1:n) {
y[i] ~ dnorm(mu, tau)
}


## Prior on mu

mu1 ~ dnorm(mus[1], taus[1])
mu2 ~ dnorm(mus[2], taus[2])
I   ~ dbinom(pis[1], 1) # Choose distribution 1 with probability 1.
mu = I*mu1 + (1-I)*mu2

## Prior on tau
tau ~ dgamma(a, b)

## Convert precision to variance
sig2 = 1/tau

## PREDICTIONS
for (i in 1:5){
  ypred[i]~ dnorm(mu, tau)
}

}"

# Data to feed to JAGS

data <- list(y = y, n = n,
             mus = mus, taus = taus, pis = pis,
             a   = a, b   = b)


# Compile and initialise model

model.1d <- jags.model(textConnection(mod.1d), n.chains = 5, data = data)

# Update: Select burn-in of 10000
update(model.1d, 20000)

# Get samples

res.1d <- coda.samples(model.1d,
                       variable.names = c("mu", 'ypred'),
                       n.iter = 20000)

## EXTRACT RELEVANT DATA
mus       <- res.1d[[1]][, 'mu']
ypred.mat <- as.matrix(res.1d[[1]][, -1])

## pr(mu < 1)
mean((mus < 1))
## 0.3834

## Pr(at least one in sample has negative)
neg.lc <- apply(ypred.mat, 1, FUN =function(x) sum(x<0) > 0)
prlo <-  sum(neg.lc)/length(neg.lc)
## 88%

mu.text <- intToUtf8(956)

label1 <- paste('P(', mu.text, '<1 | y)', sep = '')
label2 <- paste('P(>0 log concentrations negative | y)')

df.1d <- t(cbind(mean(mus<1), prlo))
df.1d <- signif(100*df.1d, 3)
df.1d <- apply(df.1d, 2, paste, '%', sep='')
rownames(df.1d) <- c(label1, label2)
```

```
colnames(df.1d) <- NULL
df.1d

### SAVE TABLE

png(filename = 'Q1/1dresults.png', width=300, height=40)
grid.table(df.1d)
dev.off()



## Can also be done binomially
1 - (sum(res.1d[[1]][, 'ypred[1]'] > 0)/20000)^5
## 89%
```

B Appendix: Question 2

B.1 2b

```r
prior.alpha <- 0.001
prior.beta  <- 0.001
y           <- 3
k           <- 3/12
post.alpha <- prior.alpha + y
post.beta  <- prior.beta  + k

# Mean, sd, 95\% CI
mean.2b <- post.alpha/post.beta
sd.2b   <- sqrt(post.alpha/(post.beta^2))
sci.2b <- c(qgamma(0.025, post.alpha, post.beta),
         qgamma(0.975, post.alpha, post.beta))

# Posterior plot

x   <- seq(0, mean.2b+4*sd.2b, 0.1)
f.x <- dgamma(x, post.alpha, rate = post.beta)
xlabel <- intToUtf8(955)
ylabel <- paste('p(', xlabel, '| y)', sep = '')
sigma  <- intToUtf8(963)



png('Q2/2bplot.png', height = 350, width = 500)
plot(x, f.x, 'l',
     xlab = xlabel, ylab = ylabel, main = 'Posterior Density Plot')
abline(v = mean.2b        , col = 'red')
abline(v = sci.2b[1]      ,  col = 'green')
abline(v = sci.2b[2]      ,  col = 'green')
abline(v = mean.2b + sd.2b,  col = 'blue')
abline(v = mean.2b - sd.2b,  col = 'blue')

legend('topright',
       legend=c('Posterior', 'Mean','95% Confidence Interval', paste('Mean +-', sigma)),
       col=c('black','red','green', 'blue'),lty=1, lwd = c(2,2,2),
       cex = 0.8)
dev.off()
```

B.2 2c

```r
plambmt15.orig <- pgamma(15, post.alpha, post.beta, lower.tail = FALSE)
```

B.3 2d

```r
### SETUP ###
prior.alpha.J <- 0.5
prior.beta.J  <- 0
y             <- 3
k             <- 3/12
post.alpha.J  <- prior.alpha.J + sum(y)
post.beta.J   <- prior.beta.J  + k

### Mean, sd, 95\% CI ###

mean.2d     <- post.alpha.J/post.beta.J
sd.2d       <- sqrt(post.alpha.J/(post.beta.J^2))
sci.2d      <- c(qgamma(0.025, post.alpha.J, post.beta.J),
                 qgamma(0.975, post.alpha.J, post.beta.J))

# Posterior plot

x      <- seq(0, mean.2d+4*sd.2d, 0.1)
f.x    <- dgamma(x, post.alpha.J, post.beta.J)
xlabel <- intToUtf8(955)
ylabel <- paste('p(', xlabel, '| y)', sep = '')
sigma  <- intToUtf8(963)

### SAVE PLOT ###
```

```

png('Q2/2dplot.png', height = 350, width = 500)

### 2D
plot(x, f.x, 'l',
      xlab = xlabel, ylab = ylabel, main = 'Posterior Density Plot')
abline(v = mean.2d         ,  col = 'red')
abline(v = sci.2d[1]       ,  col = 'green')
abline(v = sci.2d[2]       ,  col = 'green')
abline(v = mean.2d + sd.2d,   col = 'blue')
abline(v = mean.2d - sd.2d,   col = 'blue')

### 2B
abline(v = mean.2b         ,  col = 'red',    lty = 3)


legend('topright',
        legend=c('Posterior', 'Mean','95% Confidence Interval', paste('Mean +-', sigma), 'Mean (2b)'),
        col=c('black','red','green', 'blue', 'red'),lty=c(1,1,1,1,3), lwd = 2,
        cex = 0.8)


dev.off()


#### 2D:C ####
plambmt15.J <- pgamma(15, post.alpha.J, post.beta.J, lower.tail = FALSE)

#### TABULAR COMPARISON
## originalr esults
orig <-rbind(mean.2b, sd.2b, sci.2b[1], sci.2b[2], plambmt15.orig)

#### Jeffreys
jeff <- rbind(mean.2d, sd.2d, sci.2d[1], sci.2d[2], plambmt15.J)

#### Overall
comp.Q2d <- cbind(orig, jeff)
lambda <- intToUtf8(955)
sigma  <- intToUtf8(963)
gteq   <- intToUtf8(8805)

rownames(comp.Q2d) <- c('Mean', 'SD', '2.5%', '97.5%',
                         paste('P(',lambda, gteq, '15)', sep=''))
colnames(comp.Q2d) <- c('Original', 'Jeffreys')

# Multiply last row by 100 to get as percentage
comp.Q2d[5, ] <- 100*comp.Q2d[5, ]
# 4 sf for whole matrix
comp.Q2d <- signif(comp.Q2d, 4)
# Add % on the end to show percentage
comp.Q2d[5, ]  <- sapply(comp.Q2d[5,], paste, '%', sep='')
# Transpose to take up less space
comp.Q2d <- t(comp.Q2d)

png('Q2/2dComp.png', height=100, width = 300)
grid.table(comp.Q2d)
dev.off()
```

B.4 2e

```
gp.nbinom <- function(ypred, kpred, post.alpha, post.beta, lower.tail = TRUE) {
  #
  # Adapts nbinom to work with gamma posterior parameters.
  #
  # ypred:       P(X <= ypred) by default
  # kpred:       Conversion parameter associated with ypred
  # post.alpha:  Posterior from previous
  # post.beta:   Posterior from previous
  # lower.tail:  P(X <= ypred) default

  r <- post.alpha
  p <- post.beta/(kpred + post.beta)

  pnbinom(ypred, size=r, prob=p, lower.tail = lower.tail)
}
```

```r

### Various ys and ks (note, vary lower.tail, so not exact)

ypred.6m <- 5; kpred.6m <- 6/12 ## 6 or more  in 6m
ypred.5m <- 1; kpred.5m <- 5/12 ## 1 or fewer in 5m
ypred.1m <- 4; kpred.1m <- 1/12 ## 5 or more  in 1m.
## 5 or more in at least one month
## define p = P(5 or more in 1m)
# (6 choose 0) * p^0 *(1-p)^6 is P(4 or less in all months)
# Do 1 - this.



####### ORIGINAL PRIOR ######
r <- post.alpha

## 6 or more in O2
sixplus.orig <- gen.nbinom(ypred.6m, kpred.6m,
                           post.alpha=post.alpha, post.beta = post.beta,
                           lower.tail = FALSE)

## 1 or fewer in first 5 months: P(ypred <= 1)
oneminus.orig <- gen.nbinom(ypred.5m, kpred.5m,
                            post.alpha=post.alpha, post.beta = post.beta,
                            lower.tail = TRUE)

## 5 or more in last month: P(ypred >= 5) = 1- P(ypred <=4)
fiveplus.orig <- gen.nbinom(ypred.1m, kpred.1m,
                            post.alpha = post.alpha, post.beta=post.beta,
                            lower.tail = FALSE)

# 5 or more in at least one month
fourless.all.orig <- 1 - (1-fiveplus.orig)^6


###### JEFFREYS PRIOR ######

r <- post.alpha.J

## 6 or more in O2
sixplus.J <- gen.nbinom(ypred.6m, kpred.6m,
                           post.alpha=post.alpha.J, post.beta = post.beta.J,
                           lower.tail = FALSE)

## 1 or fewer in first 5 months: P(ypred <= 1)
oneminus.J <- gen.nbinom(ypred.5m, kpred.5m,
                            post.alpha=post.alpha.J, post.beta = post.beta.J,
                            lower.tail = TRUE)

## 5 or more in last month: P(ypred >= 5) = 1- P(ypred <=4)
fiveplus.J <- gen.nbinom(ypred.1m, kpred.1m,
                            post.alpha = post.alpha.J, post.beta=post.beta.J,
                            lower.tail = FALSE)

# 5 or more in at least one month
fourless.all.J <- 1 - (1-fiveplus.J)^6


### Combine results
origres <- cbind(sixplus.orig, oneminus.orig, fiveplus.orig, fourless.all.orig)
jeffres <- cbind(sixplus.J, oneminus.J, fiveplus.J, fourless.all.J)

allres <- rbind(origres, jeffres)
rownames(allres) <- c('Original', 'Jeffreys')
colnames(allres) <- c('>=6 (6m)', '<=1 (First 5m)', '>=5 (Last Month)', '>=5 (Any month)')

allres <- signif(100*allres, 4)
allres <- apply(allres, c(1,2), paste, '%', sep='')

### Write results
png('Q2/2eJeffOrig.png', height =90, width = 420)
grid.table(allres)
dev.off()
```

C Appendix: Question 3

C.1 EDA

```r
## Load Data

library(rjags)
library(e1071)
library(gridExtra)


data("rock")
area  <- rock$area
peri  <- rock$peri
perm  <- rock$perm
shape <- rock$shape

#### Define Functions ####

#### Plotting ####


data.plots <- function(yname, xnames, data, mains = c('')) {
  i <- 0
  for (name in xnames) {
    i = i + 1

    scatter.smooth(data[, yname] ~ data[, name],xlab=name, ylab = '',main=mains[i])
   # plot(data[, name], data[, yname], xlab = name, ylab = '', main = mains[i])
    #abline(lm(data[, yname] ~ data[, name]))
  }
}

#### Convergence ####

#### ESS
ESSplot <- function(sample.obj) {
  ESS <- data.frame(effectiveSize(sample.obj[[1]]))
  ESS <- data.frame(variable = row.names(ESS), ESS, row.names = NULL)
  colnames(ESS) <- c('Variable', 'ESS')
  ggplot(data = ESS, aes(x = Variable, y = ESS)) +
    geom_bar(stat = 'Identity') +
    ggtitle('Effective Sample Size for Variables')
}
#### GELMAN.DIAG
gelman.diag.df <- function(sample){

  a <- gelman.diag(sample)
  a <- round(a$psrf, 2)
  grid.table(a)

}

## ==============================================================================##
##################################### RESIDUALS###################################
## ==============================================================================##

fitted.blr <- function(resmat, X, B, sigma  = NULL, resids = FALSE){
  # resmat : Results matrix (e.g. as.matrix(samp[[1]]))
  # X      : X matrix
  # B      : Beta matrix
  # sigma  : Standard deviation (only needed for resids = TRUE)
  # resids : If true, return studentised residuals.

  # Observations and number of iterations
  n <- nrow(X)
  niterf <- nrow(resmat)

  # Get individual vectors
  b1    <- B[, 1]
  b2    <- B[, 2]
  b3    <- B[, 3]
  b4    <- B[, 4]
  # Hat matrix
  H <- X %*% solve(t(X) %*% X) %*% t(X)

  # Y = XB
  fittedvalues <- X%*% t(cbind(b1, b2, b3, b4))

  if (resids == TRUE) {

    studentisedresid=matrix(0, nrow=n, ncol=niterf)

    # For each posterior sample
    for(l in 1:niterf){
      # For each observation
      for(i in 1:n){
        # Residual/sigma *
        studentisedresid[i,l]=(l.perm[i]-fittedvalues[i,l])/(sigma[l]*sqrt((1-diag(H)[i])))
      }
    }

    return(studentisedresid)
  } else {
    return(fittedvalues)
  }
}

## ========================================================================##
#################################### SCALING ##################################
## ========================================================================##

standardise <- function(x) {(x - mean(x))/sd(x)}

## ========================================================================##
################################## EXPLORATORY ################################
## ========================================================================##

#### Univariate ####


png('Q3/3aEDAHist.png', height = 400, width = 600)
par(mfrow = c(2,2))
hist(perm,  main = 'Permeability Histogram', xlab = 'Permeability')
hist(area,  main = 'Area Histogram',         xlab = 'Area')
hist(peri,  main = 'Perimeter Histogram',    xlab = 'Perimeter')
hist(shape, main = 'Shape Histogram',        xlab = 'Shape')
par(mfrow = c(1,1))
dev.off()

#### Transformations ####

l.perm      <- log(perm)
area.norm   <- standardise(area)
peri.norm   <- standardise(peri)
shape.norm  <- standardise(shape)

rock.t <- as.data.frame(cbind(area.norm, peri.norm, shape.norm,l.perm))

#### Bivariate ####

png('Q3/3aEDABiv.png', height = 400, width = 600)
par(mfrow=c(2,2))
data.plots('l.perm', c('peri.norm', 'area.norm', 'shape.norm'), data=rock.t,
           mains = c('Permeability vs Perimeter', 'Permeability vs Area',
                     'Permeability vs Shape'))
par(mfrow=c(1,1))
dev.off()

#### Frequentist ####

freq <- lm(l.perm ~ peri.norm + area.norm + shape.norm, data = rock.t)

#### Results:
png('Q3/3aEDAfreqcoef.png', height=102, width = 310)
grid.table(round(summary(freq)$coef, 3))
dev.off()


### Save residual plot
```

```
png('Q3/3aEDAfreqres.png', height = 300, width = 1200)
par(mfrow = c(1, 4))
plot(freq)
par(mfrow = c(1,1))
dev.off()
```

C.2 Model and Convergence

```
## Define hyperparameters
beta.mu.prior   <- rep(0, 4)
beta.tau.prior  <- diag(1/100000, 4)
tau.shape.prior <- 0.1
tau.rate.prior  <- 0.1


## Data defining
x     <- cbind(1, rock.t$peri.norm, rock.t$area.norm, rock.t$shape.norm)
y     <- rock.t$l.perm
n     <- nrow(x)

## Data list
data <- list(y = y, x = x, n = n, # The dataset
             beta.mu.prior   = beta.mu.prior,
             beta.tau.prior  = beta.tau.prior,
             tau.shape.prior = tau.shape.prior, tau.rate.prior = tau.rate.prior)

model_string <-
  "model {

# Likelihood
for (i in 1:n) {
y[i] ~ dt(mu[i], tau, nu)
mu[i] = inprod(beta[], x[i, ])
}

# Prior defined on vector
beta ~ dmnorm(beta.mu.prior, beta.tau.prior)

## Prior on tau
tau ~ dgamma(tau.shape.prior, tau.rate.prior)

## Prior on nu (Fixed value now, will change later)
nu = 5

## Convert precision to variance
sigma2 <- 1/tau

}"

model <- jags.model(textConnection(model_string),
                    data = data, n.chains = 3)

update(model, 100000)

samp <- coda.samples(model = model, variable.names = c('beta', 'sigma2'), n.iter = 250000, thin = 50)


## Result Summary
summary(samp)

## Save BGR plots
png('Q3/3bGelman.png', height = 600, width = 700)
gelman.plot(samp)
dev.off()


## Save trace plots

png('Q3/3bTrace.png', height = 600, width = 700)
par(mfrow = c(3, 2))
traceplot(samp)
dev.off()

## Save autocorr plots

```

```
png('Q3/3bAuto.png', height = 600, width = 700)
par(mfrow = c(3, 2))
autocorr.plot(samp[[1]])
dev.off()

## Effective Sample Size

effectiveSize(samp[[1]])

## Bayes-Freq comparison

# Keep estimate and p-value
freq.mat <- summary(freq)$coef[, c(1,2,4)]

# Bayes
b.mean <- apply(samp[[1]], 2, mean)[1:4]
b.sd   <- apply(samp[[1]], 2, sd)[1:4]
b.CI <- apply(samp[[1]],2,  quantile, probs = c(0.025, 0.975))[,1:4]
bayes.mat <- rbind(b.mean, b.sd, b.CI)
bayes.mat <- t(bayes.mat)

combined.mat <- round(cbind(freq.mat, bayes.mat),3)
colnames(combined.mat) <- c('Classical Est', 'Classical SE', 'p-value', 'Bayes Mean', 'Bayes SD', 'Bayes 2.5%', 'Bayes 97.5%')

## SAVE
png('Q3/3bBayesFreqComp.png', width = 600, height = 100)
grid.table(combined.mat)
dev.off()
```

C.3 Sensitivity

```
## Define hyperparameters
beta.mu.prior   <- rep(0, 4)
beta.tau.prior  <- diag(1/1000, 4)
tau.lower       <- 0
tau.upper       <- 100000
nu.prior        <- 0.1

## Data defining
x     <- cbind(1, rock.t$peri.norm, rock.t$area.norm, rock.t$shape.norm)
y     <- rock.t$l.perm
n     <- nrow(x)

## Data list
data <- list(y = y, x = x, n = n, # The dataset
             beta.mu.prior   = beta.mu.prior,
             beta.tau.prior  = beta.tau.prior,
             tau.lower = tau.lower, tau.upper = tau.upper,
             nu.prior = nu.prior)

model_string <-
  "model {

# Likelihood
for (i in 1:n) {
y[i] ~ dt(mu[i], tau, nu)
mu[i] = inprod(beta[], x[i, ])
}

# Prior defined on vector
beta ~ dmnorm(beta.mu.prior, beta.tau.prior)

## Prior on tau
tau ~ dunif(tau.lower, tau.upper)

## Prior on nu (Fixed value now, will change later)
nu ~ dgamma(nu.prior, nu.prior)

## Convert precision to variance
sigma2 <- 1/tau

}"

model <- jags.model(textConnection(model_string),
                     data = data, n.chains = 3)

update(model, 100000)

samp.sens <- coda.samples(model = model, variable.names = c('beta', 'sigma2', 'nu'), n.iter = 250000, thin = 50)



## ================================================================================##
######################## SENSITIVITY DIAGNOSTIC PLOTS ##########################
## ================================================================================##

## Result Summary
summary(samp.sens)

## Save BGR plots
png('Q3/3cGelman.png', height = 600, width = 700)
gelman.plot(samp.sens)
dev.off()


## Save trace plots

png('Q3/3cTrace.png', height = 600, width = 700)
par(mfrow = c(3, 2))
traceplot(samp.sens)
dev.off()

## Save autocorr plots

png('Q3/3cAuto.png', height = 600, width = 700)
par(mfrow = c(3, 2))
autocorr.plot(samp.sens[[1]])
dev.off()

## Effective Sample Size

effectiveSize(samp.sens[[1]])


## COMPARISON TO PREVIOUS PRIOR

## PREVIOUS PRIOR: MEAN, SD, QUANTILES

previous.mean <- apply(samp[[1]], 2, mean)
previous.sd   <- apply(samp[[1]], 2, sd)
previous.qnt  <- apply(samp[[1]], 2, quantile, prob = c(0.25, 0.5, 0.75))
previous.mat  <- rbind(previous.mean, previous.sd, previous.qnt)
previous.mat  <- t(previous.mat)
colnames(previous.mat) <- c('Orig. Mean', 'Orig. Sd', '25%', '50%', '75%')
previous.mat


## NEW PRIOR
new.mean <- apply(samp.sens[[1]], 2, mean)
new.sd   <- apply(samp.sens[[1]], 2, sd)
new.qnt  <- apply(samp.sens[[1]], 2, quantile, prob = c(0.25, 0.5, 0.75))
new.mat  <- rbind(new.mean, new.sd, new.qnt)[, c(1,2,3,4,6)]
new.mat  <- t(new.mat)
colnames(new.mat) <- c('New Mean', 'New Sd', '25%', '50%', '75%')

## Comparison matrix
prior.comp.mat <- round(cbind(previous.mat, new.mat), 3)

## Save
png('Q3/3cPriorComparison.png', height = 150, width = 605)
grid.table(prior.comp.mat)
dev.off()

## ================================================================================##
############################## SENSITIVITY (NORMAL) ##############################
## ================================================================================##

## Define hyperparameters
beta.mu.prior   <- rep(0, 4)
beta.tau.prior  <- diag(1/1000, 4)
tau.lower       <- 0
tau.upper       <- 100000


## Data defining
x     <- cbind(1, rock.t$peri.norm, rock.t$area.norm, rock.t$shape.norm)
y     <- rock.t$l.perm
n     <- nrow(x)

## Data list
data <- list(y = y, x = x, n = n, # The dataset
             beta.mu.prior   = beta.mu.prior,
             beta.tau.prior  = beta.tau.prior,
             tau.lower = tau.lower, tau.upper = tau.upper)

model_string <-
  "model {

# Likelihood
for (i in 1:n) {
y[i] ~ dnorm(mu[i], tau)
mu[i] = inprod(beta[], x[i, ])
}

# Prior defined on vector
beta ~ dmnorm(beta.mu.prior, beta.tau.prior)

## Prior on tau
tau ~ dunif(tau.lower, tau.upper)

## Convert precision to variance
sigma2 <- 1/tau

}"

model <- jags.model(textConnection(model_string),
                    data = data, n.chains = 3)

update(model, 100000)

samp.sens.norm <- coda.samples(model = model, variable.names = c('beta', 'sigma2'), n.iter = 250000, thin = 50)

##### NORMAL DISTRIBUTION COMPARISON

## PREVIOUS PRIOR: MEAN, SD, QUANTILES

previous.mean <- apply(samp[[1]], 2, mean)
previous.sd   <- apply(samp[[1]], 2, sd)
previous.qnt  <- apply(samp[[1]], 2, quantile, prob = c(0.25, 0.5, 0.75))
previous.mat  <- rbind(previous.mean, previous.sd, previous.qnt)
previous.mat  <- t(previous.mat)
colnames(previous.mat) <- c('Orig. Mean', 'Orig. Sd', '25%', '50%', '75%')
previous.mat


## NEW PRIOR
new.mean.norm <- apply(samp.sens.norm[[1]], 2, mean)
new.sd.norm   <- apply(samp.sens.norm[[1]], 2, sd)
new.qnt.norm  <- apply(samp.sens.norm[[1]], 2, quantile, prob = c(0.25, 0.5, 0.75))
new.mat.norm  <- rbind(new.mean, new.sd, new.qnt)
new.mat.norm  <- t(new.mat)
colnames(new.mat.norm) <- c('New Mean', 'New Sd', '25%', '50%', '75%')

## Comparison matrix
prior.comp.mat <- round(cbind(previous.mat, new.mat), 3)

## Save
png('Q3/3cPriorComparisonNormal.png', height = 150, width = 605)
grid.table(prior.comp.mat)
dev.off()
```

C.4 Model Checks

```r
## ==========================================================================##
########################## STUDENTISED RESIDUALS ################################
## ==========================================================================##

# Use one chain for the results

resmat <- as.matrix(samp[[1]])

# Individual parameter vectors
b1    <- resmat[, 1]
b2    <- resmat[, 2]
b3    <- resmat[, 3]
b4    <- resmat[, 4]
sigma <- sqrt(resmat[, 5])

# beta matrix
B <- cbind(b1, b2, b3, b4)

# residuals and fitted
studentisedresid <- fitted.blr(resmat = resmat, X = x, B = B,
                               sigma = sigma, resids = TRUE)
fittedvalues     <- fitted.blr(resmat = resmat, X = x, B = B,
                               sigma = NULL, resids = FALSE)

## ==========================================================================##
############################ RESIDUAL ANALYSIS ##################################
## ==========================================================================##

# posterior mean of studentised residuals
# Get mean of each row, so we have 48 residuals

studentisedresidm <- apply(studentisedresid, 1, mean)

## Index plot

png('Q3/3eresidindex.png', height = 350, width = 500)
plot(studentisedresidm, main = 'Residuals vs Index')
dev.off()

#QQ-plot

png('Q3/3eQQplot.png', height=350, width=500)
qqplot(x=qt(ppoints(n), df=5),
       y=studentisedresidm,
       main="QQ Plot (T- Distribution)",
       xlab="Theoretical Quantiles",
       ylab= "Sample Quantiles")
qqline(studentisedresidm, distribution=function(p) qt(p, df=5), col="red", lw=2)
dev.off()



######## PREDICTIVE CHECKS
######## REPLICATE DATA

#Now do some predictive checks
#First replicate the data


niterf <- nrow(resmat)

yrep=matrix(0,nrow=n,ncol=niterf)
for(l in 1:niterf){
  for(i in 1:n){
    yrep[i,l]=rnorm(1,b1[l]*x[i,1]+b2[l]*x[i,2]+b3[l]*x[i,3]+b4[l]*x[i,4],sigma[l])
  }
}

#Compute posterior preditive distribution of important stats
yrepmin  = apply(yrep,2,min)
yrepmax  = apply(yrep,2,max)
yreplow  = apply(yrep,2,quantile, probs = c(0.25))
yrepmed  = apply(yrep,2,median)
yrepupp  = apply(yrep,2,quantile, probs = c(0.75))
ypredkrt = apply(yrep, 2, kurtosis)

```

```r

png('Q3/3ePostPredChecks.png', height=500, width=700)
par(mfrow = c(2, 3))
##### Posterior histograms
hist(yrepmin, col = 'gray40', main = 'Predictive Distribution for Minimum')
  abline(v = min(l.perm), col = 'red', lwd = 2)

hist(yrepmax, col = 'gray40', main = 'Predictive Distribution for Maximum')
  abline(v = max(l.perm), col = 'red', lwd = 2)

hist(yreplow, col = 'gray40', main = 'Predictive Distribution for Lower Quartile')
  abline(v = quantile(l.perm, probs = c(0.25)), col = 'red', lwd = 2)

hist(yrepmed, col = 'gray40', main = 'Predictive Distribution for Median')
  abline(v = median(l.perm), col = 'red', lwd = 2)

hist(yrepupp, col = 'gray40', main = 'Predictive Distribution for Upper Quartile')
  abline(v = quantile(l.perm, probs = c(0.75)), col = 'red', lwd = 2)

hist(ypredkrt, col = 'gray40', main = 'Predictive Distribution for Kurtosis')
  abline(v = kurtosis(l.perm), col = 'red', lwd = 2)
dev.off()
```