1. Beam Dataset Analysis- Bayesian Linear Regression

For the Beam Dataset acquired from Abaqus, Bayesian Linear Regression was performed. The displacement data collection will be discussed first.

Displacement Data

Displacement_Values	Log Displ Values	average	std deviation
0.88	0.631272	1.328385	0.762252
0.76	0.565314		
1.16	0.770108		
0.97	0.678034		
1.11	0.746688		

Figure 1 Preview of Displacement variable data metrics

Before running the Bayesian Linear Regression model on the displacement dataset, the hyperparameters (Prior Mean and Standard Deviation, Prior Uncertainty for regression coefficients, and overall data noise level) had to be determined. A prior predictive check was performed as an attempt to fine tune the parameters. From the initial assumptions of the hyperparameters, sample outcomes are generated and compared against the actual data. If the results don't match the trend or scale of the actual data, the hyperparameters are tuned further. The hyperparameters were not adjusted to perfectly match the actual data but were inclined to match the overall trend and scale.

The initial Prior Mean and Standard Deviation were directly calculated from the log transformed displacement data. Then the initial prior uncertainty and overall data noise level had coefficient values of 1.0 and 0.76 respectively (identical to the Standard Deviation of the displacement dataset values).

Utilizing the prior predictive checking method, a histogram of the simulated log transformed displacement data was displayed alongside with the actual observed log transformed data to compare. Adjustments to the coefficients were made (the Prior Mean and Standard Deviation were close to the original values) so that the overall trend and values are close to the observed data.

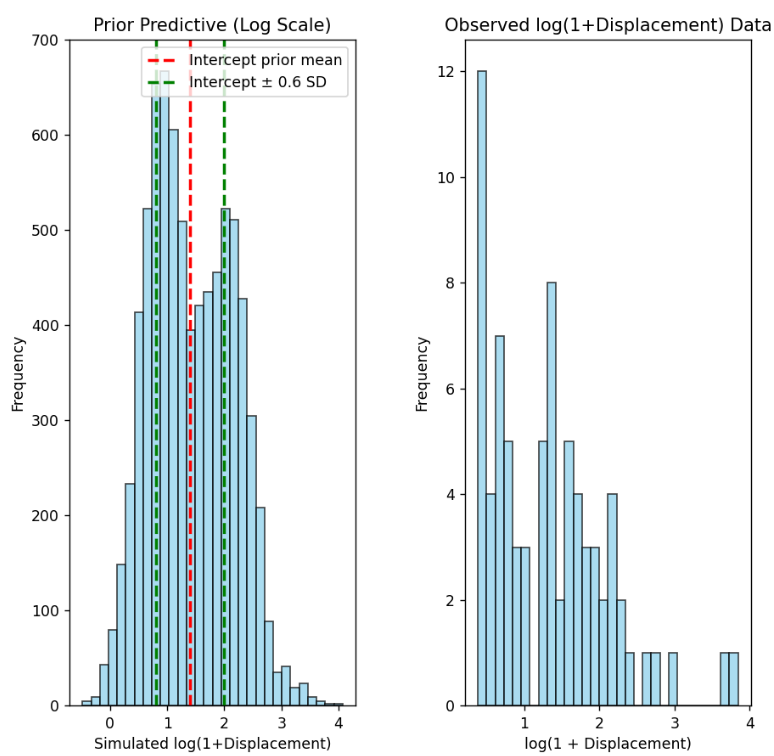


Figure 2 Prior Predictive comparison of Priors and Observed data

Then, the predicted log scale displacement values were transformed back to the original scale and compared to the raw displacement data. The overall trends and scale of the distribution were also considered when adjusting the priors.

Once the scale and distribution of the data based on the priors were deemed acceptable, Bayesian Linear Regression was performed on the dataset. The displacement raw data was log transformed, and the independent variables were Standardized, like Ridge and Lasso Regression methods in the first homework.

```
Bayesian Regression Performance on Test Set (log-transformed model):
MSE: 3.2261164931717596
MAE: 1.1457567733028613
R2 Score: 0.7992000779940878
```

Figure 4 Displacement Bayesian Regression result

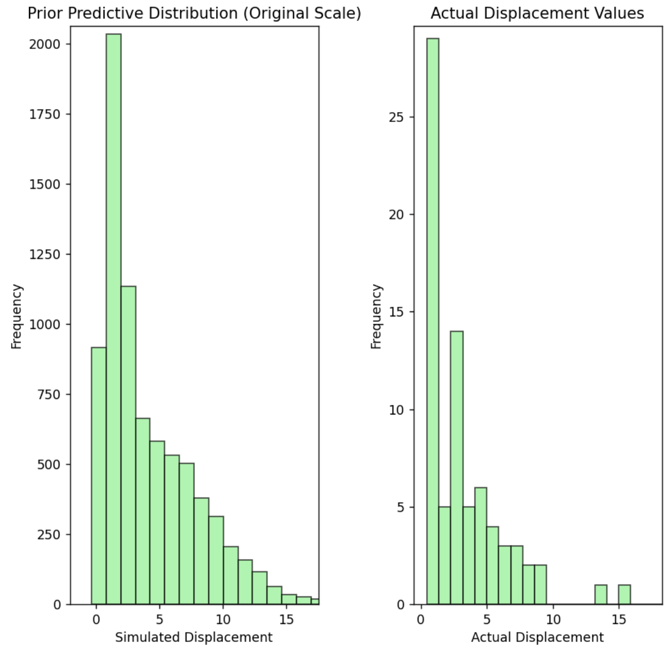


Figure 3 Prior predictive result of actual values vs prior generated results

As shown above, the performance of Bayesian Regression is a little better, seeing that the best performance from the Ridge/Lasso Regression runs had an R squared value of 0.7812 with MAE of 1.16 and MSE of 3.51 and Bayesian Regression had R squared value of 0.799, MAE of 1.14 and MSE of 3.22.

The Actual vs Observed plot demonstrates the needed for more runs with higher Stress values, to potentially improve the model given that the stress values were more scattered around the upper Stress region but the lower Stress values were more concentrated around the ideal line. This also may indicate that there may be unrealistic phenomena occurring at higher stress levels (such as voids being too close to the edge of the beam).

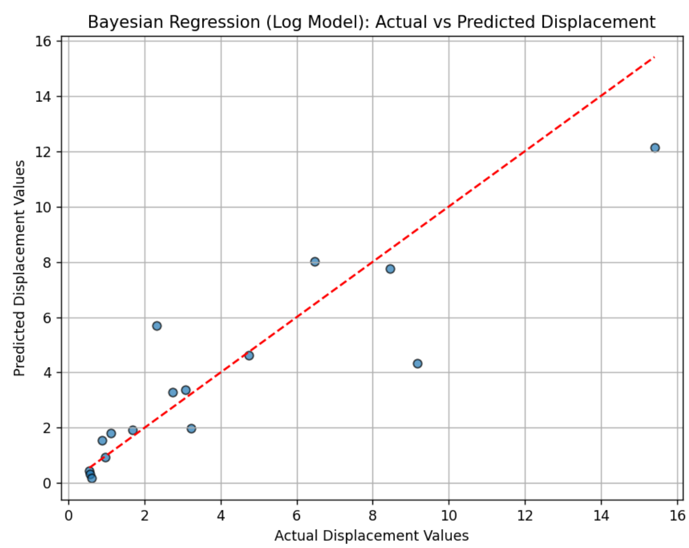


Figure 5 Actual vs Predicted Displacement values for Beam data

Sampling 4 chains for 1_000 tune and 2_000 draw iterations (4_000 + 8_000 draws total)

	mean	sd	hdi_3%	hdi_97%	mcse_mean	mcse_sd	ess_bulk	ess_tail	r_hat
intercept	1.32	0.07	1.18	1.46	0.00	0.00	4039.73	5368.12	1.0
coefs[0]	-0.17	0.35	-0.85	0.44	0.01	0.00	3779.59	4781.90	1.0
coefs[1]	1.10	0.08	0.95	1.24	0.00	0.00	8053.17	5784.87	1.0
coefs[2]	-0.38	0.11	-0.59	-0.18	0.00	0.00	5548.44	5434.95	1.0
coefs[3]	-0.48	0.12	-0.70	-0.27	0.00	0.00	5410.16	5608.04	1.0
coefs[4]	0.11	0.17	-0.20	0.44	0.00	0.00	3517.61	4295.72	1.0
coefs[5]	0.45	0.40	-0.29	1.24	0.01	0.01	3211.63	4494.90	1.0
coefs[6]	0.08	0.08	-0.07	0.24	0.00	0.00	3886.18	5046.38	1.0
coefs[7]	-0.15	0.12	-0.39	0.07	0.00	0.00	6335.14	5805.21	1.0
coefs[8]	-0.09	0.14	-0.34	0.18	0.00	0.00	4096.11	4916.89	1.0
sigma	0.31	0.03	0.26	0.36	0.00	0.00	6684.19	5817.86	1.0

Figure 6 Convergence table results for each independent variable

The trace plot is displayed to the right. Here, it shows the Markov Chain Monte Carlo parameter values on the right and Posterior Distributions to the left.

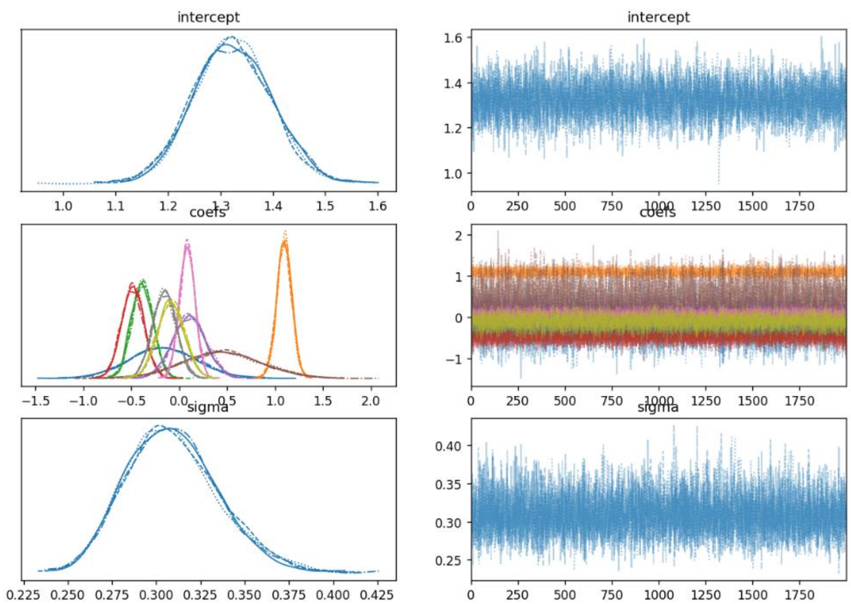


Figure 7 Trace plots for displacement beam data

The intercept plot peaks around 1.35, which is near the original prior value of 1.4.

Most of the coefficient distribution is narrow, with some values near 0, and some values with a relatively large distribution indicating there's some uncertainty about some coefficients or the prior is vague. However, it appears most of the coefficients has a negative inclination, meaning the model is picking up that there is an inverse relationship between the responding variable and the independent variable, which is to be expected.

The sigma plot shows a near normal distribution centered around 0.3, but there is a slight right tail, but the peak is still distinct.

The plot to the left is an expansion of the coefficient plot, detailing the posterior distribution of each coefficient. Again, showing in depth that most of the coefficients have been determined to have a negative inclination on the response variable.

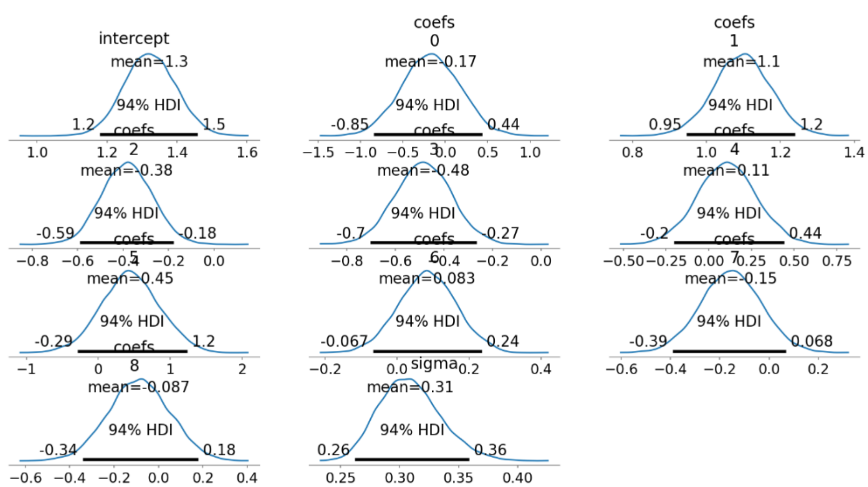


Figure 8 Coefficient Plot expanded

As a comparison, the original input coefficients for this run had the intercept mean as 1.4, intercept mean variance as 0.3, coefficient sigma as 0.60, and overall dataset variation as 0.10.

Stress Data

For the Stress Data approach, a very similar approach to the Displacement Data run was performed.

A Prior Predictive check was performed, tuning the parameters to match the overall trend of the original data.

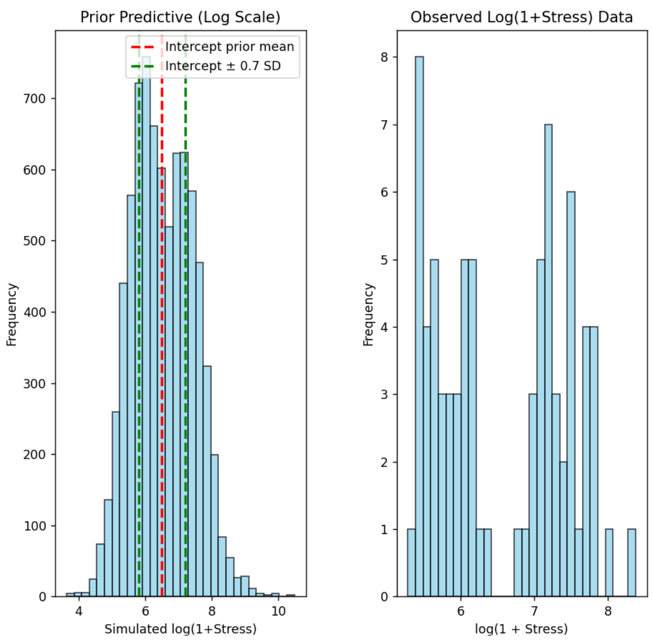


Figure 10 Prior Predictive stress distribution for simulated stress vs actual stress

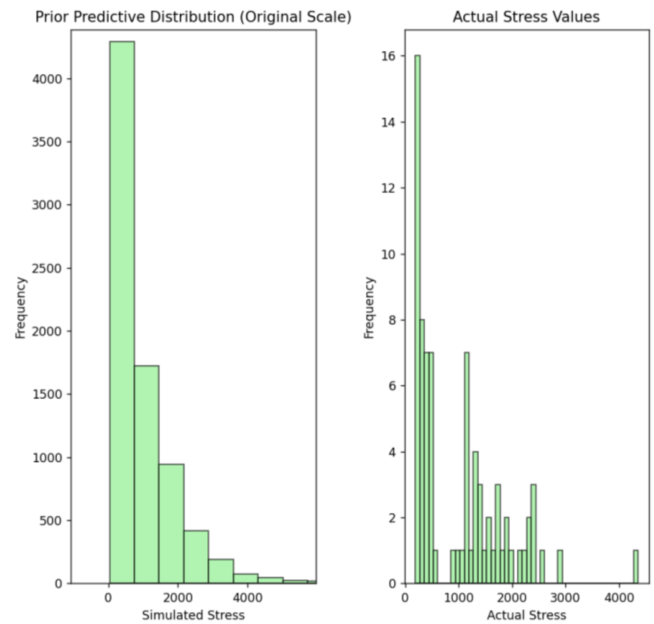


Figure 9 Displaying comparison of simulated Stress and actual Stress distribution

Again, the coefficient values had initial starting coefficients based off the raw data, primarily for the intercept mean/variance and the other coefficients were fine-tuned via the Prior Predicive.

After performing Bayesian Linear Regression on the data, the R squared value performed similarly to Lasso Regression, determined from the previous assignment.

Stress_Values	Log Stress Values	average	std deviation
447.3	6.105462651	6.585698	0.858651245
270.9	5.605434352		
513.7	6.243584207		

Figure 11 Preview of Stress variable data metrics

For comparison, the Lasso Regression model had an R squared value of 0.9072, MAE value of 149.68 and MSE value of 43690.

```
Bayesian Regression Performance on Test Set (log-transformed model)
MSE: 50830.69707304491
MAE: 164.24463652071023
R2 Score: 0.8920227869054825
```

Figure 12 Stress Bayesian Regression results

The Bayesian Linear Regression model had an R squared value of 0.89, MAE Value of 164.24, and MSE value of 50830.67.

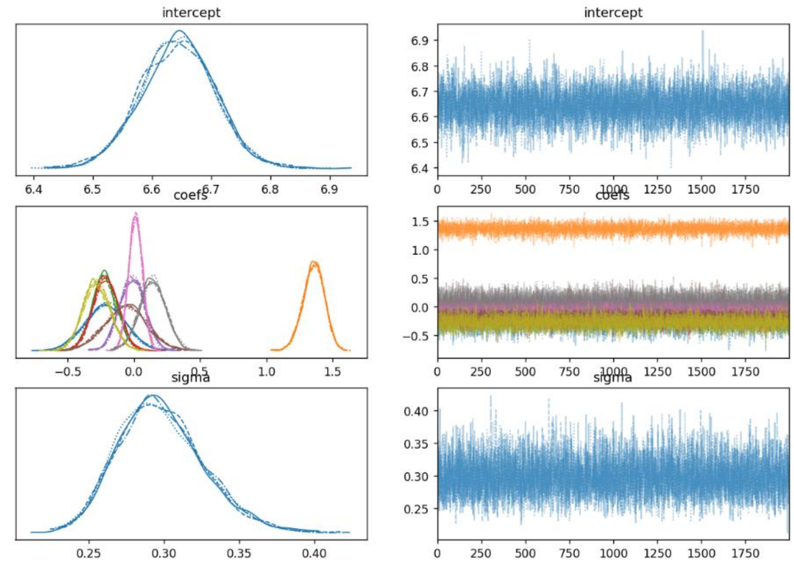


Figure 14 Trace plots for stress beam data

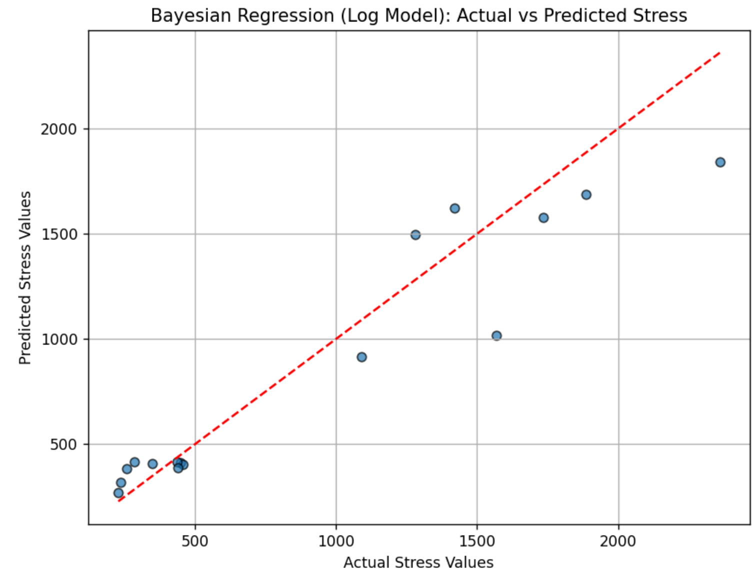


Figure 13 Actual vs Predicted Stress values for Beam data

Again, the Predicted vs Observed plot demonstrated that the lower Stress values were concentrated around the ideal line, but the higher Stress values were more scattered. This is indicative that more datapoints is needed for higher stress values to fully capture a better model.

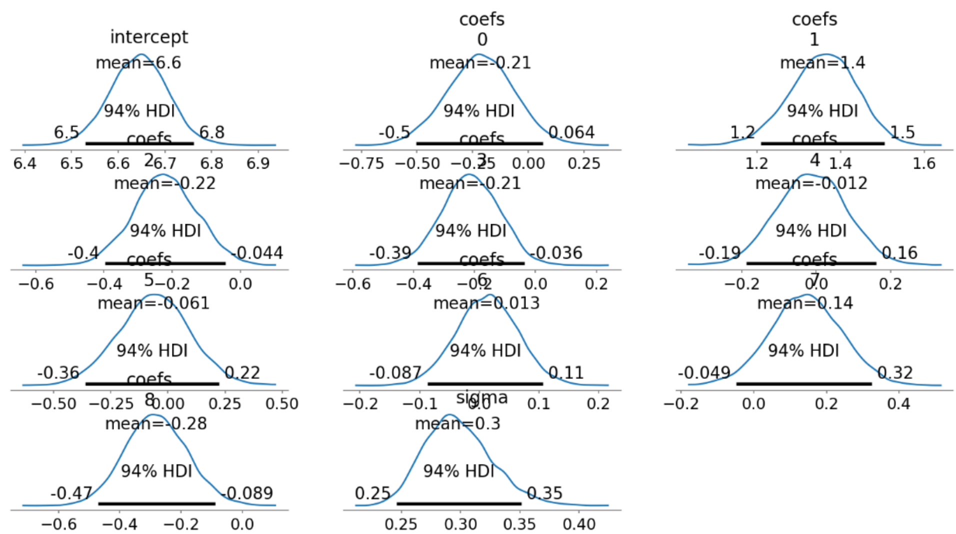


Figure 15 Expansion of coefficient plots

For the trace plots, they exhibit similar performance as the Displacement data. The original input coefficients for this run had the intercept mean as 6.5, intercept mean variance as 0.7, coefficient sigma as 0.20, and overall dataset variation as 0.10.

The Posterior priors were updated to have an intercept mean around 6.65, and overall dataset variance of around 0.3, so there were some adjustments made for the Posterior.

Expanding on the coefficients plot again, they demonstrate a very similar trend to the Displacement plots, however, most of the coefficients has a sharper distribution peak, with none having a large distribution, indicating the model has a higher confidence for all the coefficients; more than the Displacement run coefficients.

Sampling 4 chains for 1_000 tune and 2_000 draw iterations (4_000 + 8_000 draws total)

	mean	sd	hdi_3%	hdi_97%	mcse_mean	mcse_sd	ess_bulk	ess_tail	r_hat
intercept	6.64	0.06	6.53	6.76	0.0	0.0	4612.76	4857.42	1.0
coefs[0]	-0.21	0.15	-0.50	0.06	0.0	0.0	5834.86	5673.11	1.0
coefs[1]	1.36	0.08	1.21	1.50	0.0	0.0	6069.73	4521.50	1.0
coefs[2]	-0.22	0.09	-0.40	-0.04	0.0	0.0	5277.87	5674.98	1.0
coefs[3]	-0.21	0.09	-0.39	-0.04	0.0	0.0	5834.38	6139.54	1.0
coefs[4]	-0.01	0.09	-0.19	0.16	0.0	0.0	5293.16	5853.30	1.0
coefs[5]	-0.06	0.16	-0.36	0.22	0.0	0.0	4966.45	4714.70	1.0
coefs[6]	0.01	0.05	-0.09	0.11	0.0	0.0	5262.09	5422.21	1.0
coefs[7]	0.14	0.10	-0.05	0.32	0.0	0.0	6670.40	6046.05	1.0
coefs[8]	-0.28	0.10	-0.47	-0.09	0.0	0.0	5109.82	5327.75	1.0
sigma	0.30	0.03	0.25	0.35	0.0	0.0	5945.19	5905.49	1.0

Figure 16 Convergence table for stress beam data run

2. External Dataset Analysis- Bayesian Linear Regression

Strain Fatigue

Similar approaches discussed above were performed for the Strain Fatigue data for the Bayesian Linear Regression. The initial priors were fine tuned to match the overall trend of the actual data, with the initial starting point as the actual Mean and Standard Deviation of the log transformed Fatigue data. The Fatigue data in the Excel sheet was already log transformed so there was no need to further transform the data.

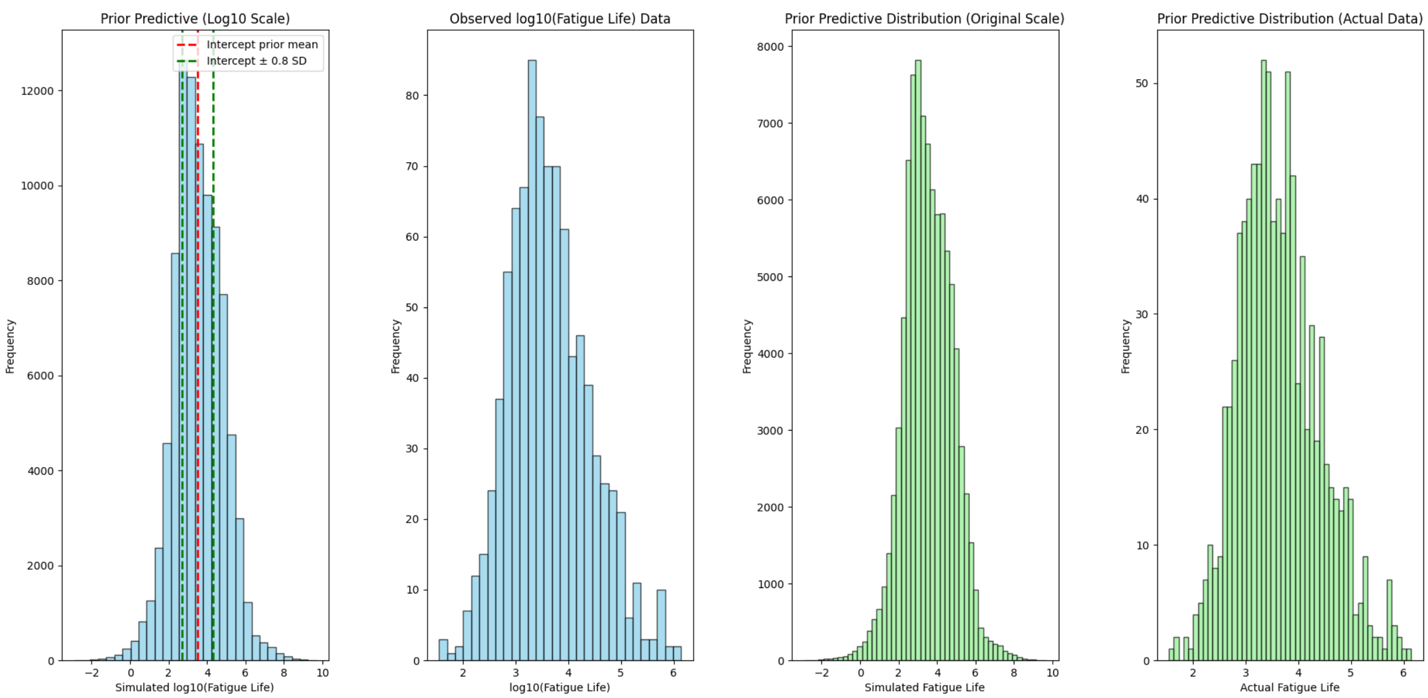


Figure 17 Prior Predictive check for Strain Fatigue data, simulated vs observed

Once the Priors were chosen, Bayesian Linear Regression was performed on the data. From Ridge/Lasso Regression in the previous assignment, a quick trade study showed that independent variables that were Standardized provided the best results. This was kept the same for the Bayesian Linear Regression runs as well to be kept consistent.

However, an interesting issue appeared where the model did not converge and provided poor results.

Here, the R squared value was 0.446, with MAE and MSE values of 0.309 and 0.350 were provided, which was the worst run yet.

A closer look at the convergence table for each coefficient shows that the R hat value was consistently above 1.01, confirming failure to converge and providing poor model results.

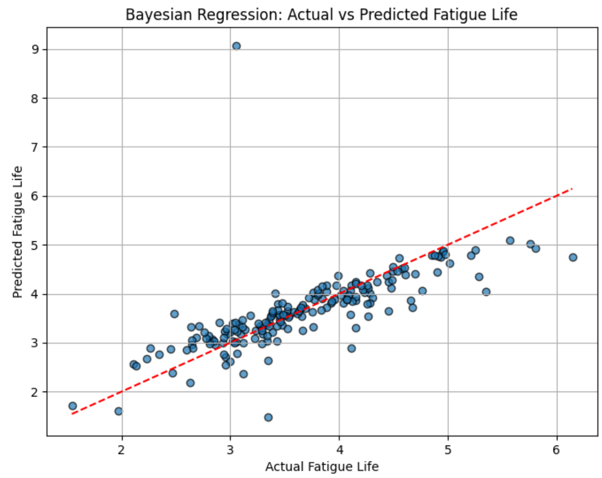


Figure 18 Actual vs Observed Strain Fatigue Data

```
[496 rows x 9 columns]
Bayesian Regression Performance on Test Set:
MSE: 0.3500268464405905
MAE: 0.30918507282630875
R2 Score: 0.44637550200753273
```

Figure 19 Bayesian Regression run results

```
Sampling 2 chains for 7_000 tune and 3_000 draw iterations (14_000 + 6_000 draws total) took 14932 seconds.0000 4:08:40<00:00 Sampling 2 chains, 0 divergences]]
We recommend running at least 4 chains for robust computation of convergence diagnostics
            mean    sd  hdi_3%  hdi_97%  mcse_mean  mcse_sd  ess_bulk  ess_tail  r_hat
intercept   0.08  0.15   -0.17     0.25       0.10     0.09      2.65     22.76   2.15
coefs[0]   -0.10  0.07   -0.19    -0.00       0.05     0.04      2.98     37.65   1.85
coefs[1]    0.61  0.01    0.60     0.62       0.01     0.01      2.63     10.82   2.12
coefs[2]   -0.44  0.01   -0.46    -0.43       0.01     0.00      2.80     21.07   2.00
coefs[3]    0.04  0.02    0.01     0.07       0.01     0.00     12.09     41.48   1.15
...          ...   ...     ...      ...        ...      ...       ...       ...    ...
coefs[490] -0.51  0.10   -0.64    -0.39       0.07     0.06      2.62     10.87   2.17
coefs[491] -0.36  0.11   -0.48    -0.19       0.08     0.07      2.35     10.90   2.65
coefs[492]  0.17  0.05    0.06     0.26       0.01     0.01     19.62     94.91   1.07
coefs[493] -0.35  0.12   -0.52    -0.18       0.08     0.07      2.63     22.85   2.15
sigma       0.37  0.01    0.35     0.38       0.00     0.00     13.11     33.85   1.12
```

Figure 20 Poor convergence table of Strain Fatigue data

The trace plots again displayed this, indicating that for this run, the model performed poorly, failed to converge to intercept, coefficient and sigma values appropriately. The MCMC results on the right plots shows erratic, unwanted behavior indicating that the model parameters must be changed for MCMC to converge.

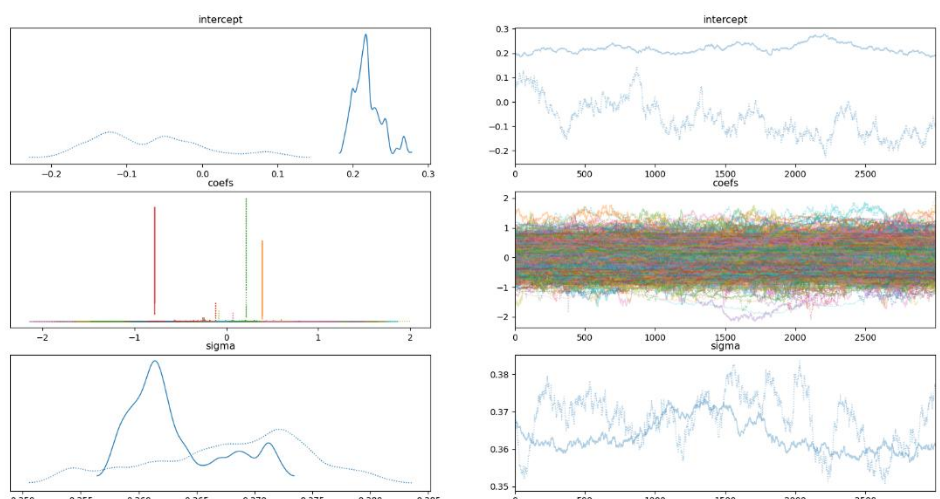


Figure 21 Trace plots for Strain Fatigue dataset

One aspect that was investigated was the number of independent variables used for this model. The Strain Fatigue dataset has the largest amount of data in any of the runs utilized. The table below displays the number of features in the model, which has a shape of (914, 488) which is very high, and after some evaluation, the Curse of Dimensionality was deemed to be the reason for the poor model performance.

Combined features shape: (914, 488)

	0	1	2	3	4	5	6	7	8	9	...	484	485	486	487	488	489	490	491	492	493
0	1.000000	0.388003	0.138276	-0.311559	-0.171946	-0.197138	-0.198244	-0.198559	-0.198761	-0.199048	...	0.001770	0.000698	0.046042	0.557853	-0.114474	-0.631574	0.067158	0.200821	-0.221716	0.302576
1	0.388003	1.000000	0.904207	-0.412656	0.106876	0.157730	0.162299	0.163978	0.164561	0.164619	...	0.064034	0.048503	0.017093	0.119688	-0.194191	-0.351332	0.475800	0.097283	0.358798	0.037208
2	0.138276	0.904207	1.000000	-0.319997	0.119550	0.162855	0.165704	0.166317	0.166221	0.165901	...	0.066313	0.054895	-0.000200	0.121850	-0.275135	-0.214625	0.483468	-0.073815	0.498317	-0.140048
3	-0.311559	-0.412656	-0.319997	1.000000	0.051829	0.048225	0.047157	0.046605	0.046352	0.046324	...	0.035575	0.018339	-0.015192	-0.251598	0.158329	0.122157	0.012408	-0.221684	-0.066053	0.127241
4	-0.171946	0.106876	0.119550	0.051829	1.000000	0.926691	0.901274	0.884035	0.870689	0.859746	...	0.176873	0.124957	-0.067312	-0.155265	0.087497	0.045176	0.052095	-0.022915	0.200150	-0.049614
..	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
489	-0.631574	-0.351332	-0.214625	0.122157	0.045176	0.043751	0.041192	0.038984	0.037104	0.035542	...	-0.041132	-0.024086	-0.028060	-0.254227	-0.092600	1.000000	-0.047248	-0.108287	-0.095495	-0.149847
490	0.067158	0.475800	0.483468	0.012408	0.052095	0.059076	0.057754	0.056120	0.054436	0.052782	...	0.024806	-0.017336	-0.011596	-0.105063	-0.038268	-0.047248	1.000000	-0.044751	-0.039465	-0.061926
491	0.200821	0.097283	-0.073815	-0.221684	-0.022915	-0.041346	-0.045556	-0.048573	-0.050943	-0.052835	...	0.036115	0.044153	0.030564	-0.240792	-0.087706	-0.108287	-0.044751	1.000000	-0.090448	-0.141928
492	-0.221716	0.358798	0.498317	-0.066053	0.200150	0.241567	0.244269	0.245172	0.245545	0.245779	...	0.040068	0.036430	-0.023438	-0.212347	-0.077345	-0.095495	-0.039465	-0.090448	1.000000	-0.125162
493	0.302576	0.037208	-0.140048	0.127241	-0.049614	-0.045956	-0.042154	-0.039096	-0.036678	-0.034860	...	0.045817	0.050225	-0.036778	-0.333207	-0.121367	-0.149847	-0.061926	-0.141928	-0.125162	1.000000

Figure 22 Original feature table prior to PCA

Principle Component Analysis (PCA) was utilized to help reduce dimensionality but keep the important variables that explains a high percentage of variability. Within the code, it was decided to keep 99% of the variance in the data. Some experimenting with values between 90% to 99% showed that 99% of the variance selection performed best.

This ensures little to no information is lost, maintaining important patterns for the model to capture, which is essential seeing that the original model had a high number of independent variables. This poses a risk of overfitting though, as it may capture noise as useful data reducing generalization. Further studies would be needed to fully understand the impact of choosing a high value but for this model it is acceptable as a first pass.

Combined features shape: (914, 488)
Reduced feature shape after PCA: (914, 81)

	0	1	2	3	4	5	6	7	8	9	...	71	72	73	74	75	76	77	78	79	80
0	1.000000	-0.017509	-0.013942	-0.041311	0.023729	-0.020959	-0.016333	-0.000276	-0.001522	0.012693	...	-0.036893	0.004244	-0.001230	-0.014318	-0.009153	-0.006656	0.017575	0.001207	0.012490	-0.007911
1	-0.017509	1.000000	-0.013397	-0.011305	0.011197	-0.020620	-0.032987	-0.000095	0.007130	0.008395	...	-0.010547	-0.008037	-0.004071	0.001299	-0.004048	0.002807	-0.001521	0.034871	0.004324	0.040065
2	-0.013942	-0.013397	1.000000	-0.015850	-0.006221	-0.003819	0.008341	-0.006388	0.000999	-0.004796	...	-0.001134	-0.009447	0.009163	0.006750	-0.002376	0.001291	0.011014	0.016914	-0.009093	0.006939
3	-0.041311	-0.011305	-0.015850	1.000000	0.023981	-0.000696	-0.002952	0.010969	-0.004720	0.024096	...	-0.026492	0.022764	0.028936	-0.038743	0.011150	0.001857	-0.011752	0.030362	0.037828	-0.028029
4	0.023729	0.011197	-0.006221	0.023981	1.000000	-0.007665	-0.004180	-0.012918	0.026574	-0.052169	...	0.028841	-0.034303	-0.010123	0.025533	-0.031676	0.004815	0.021150	-0.029299	-0.061748	0.010630
..	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
76	-0.006656	0.002807	0.001291	0.001857	0.004815	0.007719	-0.010961	-0.032969	-0.146304	-0.151518	...	0.008156	-0.011532	-0.010195	0.014542	-0.021445	1.000000	0.014237	-0.024687	-0.004158	0.015058
77	0.017575	-0.001521	0.011014	-0.011752	0.021150	-0.009502	0.071223	-0.058391	0.005528	-0.028699	...	0.056645	0.012473	-0.033533	-0.017080	0.008443	0.014237	1.000000	0.042100	-0.023963	-0.046891
78	0.001207	0.034871	0.016914	0.030362	-0.029299	0.007947	-0.007188	-0.039606	0.075555	0.008641	...	-0.012737	0.014601	0.038154	0.018193	-0.004970	-0.024687	0.042100	1.000000	-0.108007	0.052864
79	0.012490	0.004324	-0.009093	0.037828	-0.061748	0.032313	-0.082768	-0.145128	0.039498	-0.041661	...	0.020937	-0.025255	0.018769	0.001306	0.000598	-0.004158	-0.023963	-0.108007	1.000000	-0.024492
80	-0.007911	0.040065	0.006939	-0.028029	0.010630	-0.061043	-0.041315	-0.041897	-0.053521	0.014146	...	0.013854	0.028395	-0.014584	-0.006790	-0.003967	0.015058	-0.046891	0.052864	-0.024492	1.000000

Figure 23 New feature table reduced after PCA

The table above shows a significant reduction in independent variables, as the size of the features is now (914, 81) as opposed to the original scale of (914, 488).

Here, the model performed well, with an R squared score of 0.763 and MAE/MSE score of 0.272 and 0.149. Ridge and Lasso Regression models performed similarly, with Lasso Regression performing best still with an R squared value of 0.7765.

```
Bayesian Regression Performance on Test Set:
MSE: 0.14952690886636819
MAE: 0.2728908978652858
R2 Score: 0.7634988267348262
```

Figure 25 Results for Bayesian Strain Fatigue dataset

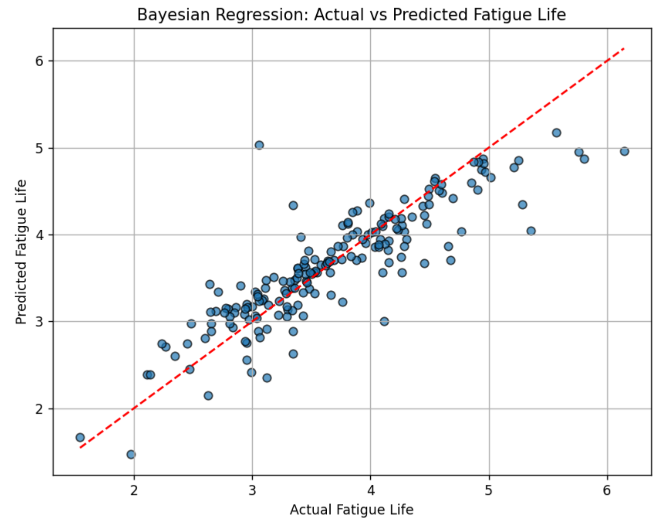


Figure 24 Improved Actual vs Predicted Strain Fatigue result

Here, the trace plots are a significant improvement on the convergence of the model on the runs that did not utilizes PCA. The original input coefficients for this run had the intercept mean as 3.5, intercept mean variance as 0.5, coefficient sigma as 0.20, and overall dataset variation as 0.10.

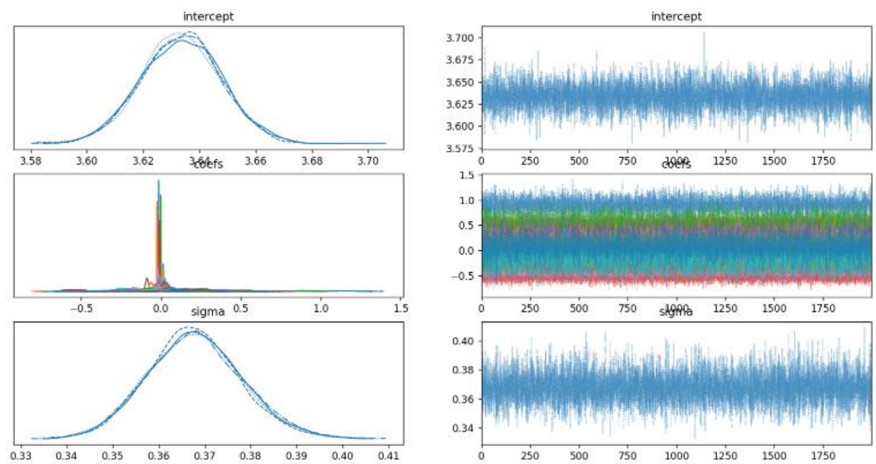


Figure 26 Trace plots for Strain Fatigue set

The Posterior priors were updated to have an intercept mean of around 3.63, and overall dataset variation near 0.37, so some adjustments were made once the actual data was included in the analysis. The short preview of the coefficient graphs also shows better convergence, as there is a single distribution present. Most of the preview coefficients were near 0 but still displayed an overall trend of having a negative relationship with the response variable, as expected.

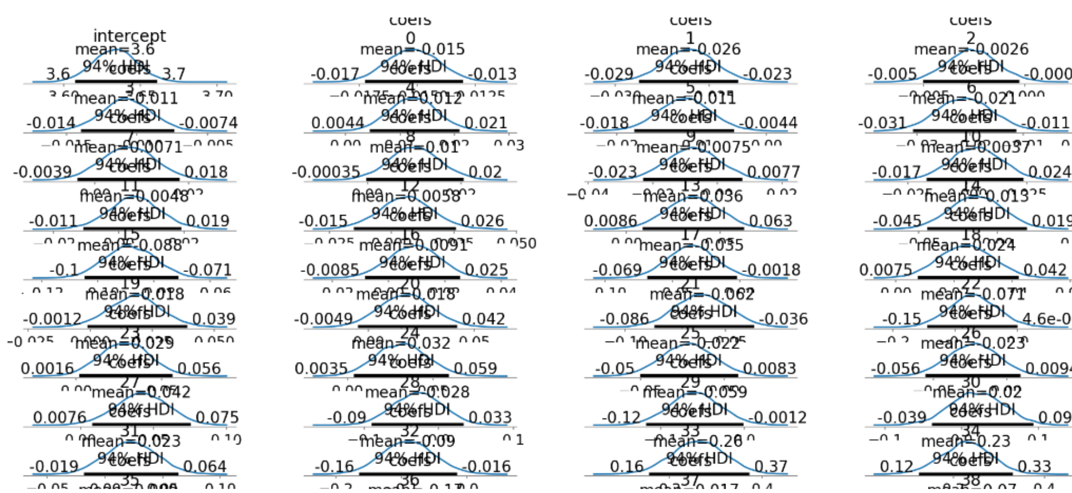


Figure 27 coefficient plot for Strain Fatigue

The table plotted to the right, further confirms better convergence as the R hat values are now 1.0, a significant improvement on the previous runs.

```
Sampling 4 chains for 4_000 tune and 2_000 draw iterations (16_000 + 8_000 draws total) took 119
             mean    sd  hdi_3%  hdi_97%  mcse_mean  mcse_sd  ess_bulk  ess_tail  r_hat
intercept    3.63  0.01    3.61     3.66        0.0      0.0   7016.40   5043.07    1.0
coefs[0]    -0.02  0.00   -0.02    -0.01        0.0      0.0   2192.34   4075.50    1.0
coefs[1]    -0.03  0.00   -0.03    -0.02        0.0      0.0   1707.41   3165.69    1.0
coefs[2]    -0.00  0.00   -0.01    -0.00        0.0      0.0   9631.40   5751.46    1.0
coefs[3]    -0.01  0.00   -0.01    -0.01        0.0      0.0   3649.65   4492.72    1.0
...           ...   ...     ...      ...        ...      ...       ...       ...    ...
coefs[77]   -0.12  0.17   -0.45     0.20        0.0      0.0  10065.07   5839.83    1.0
coefs[78]    0.05  0.17   -0.27     0.38        0.0      0.0   9234.79   5566.74    1.0
coefs[79]   -0.08  0.18   -0.43     0.25        0.0      0.0  11088.79   6043.41    1.0
coefs[80]    0.11  0.18   -0.22     0.45        0.0      0.0   8420.23   5752.57    1.0
sigma        0.37  0.01    0.35     0.39        0.0      0.0   8719.60   5208.46    1.0
```

Figure 28 Convergence table improved for Strain Fatigue post PCA

Stress Fatigue

Once again, the prior coefficients were fine-tuned and carefully selected via the Prior Predictive check and adjusted to match similar trends of the actual data.

Once the priors were chosen, Bayesian Linear Regression was performed. The same transformation of the independent data for the Strain Fatigue run was chosen, and the response variable was not transformed.

The R squared value achieved from this run was 0.357, which was a moderate improvement from the Lasso Regression run, which had an R squared value of 0.3273 but Ridge Regression had an R squared value of 0.3583 respectively.

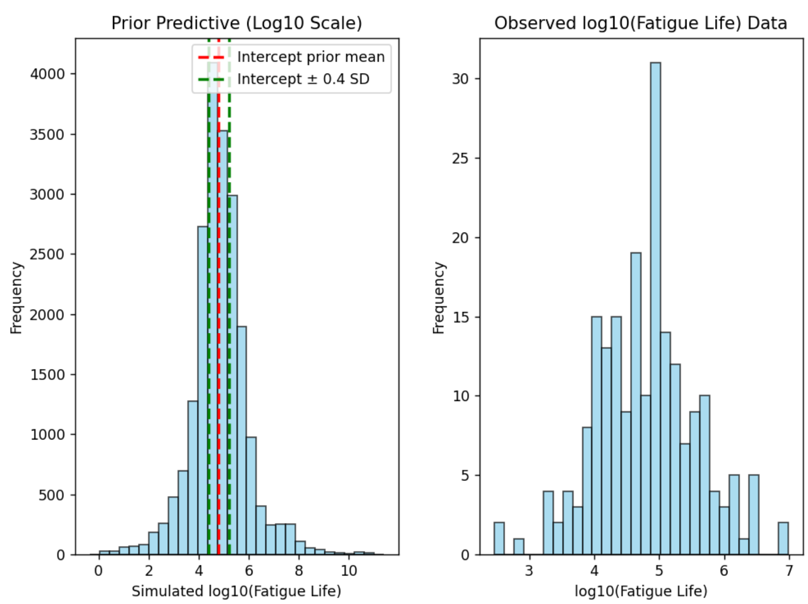


Figure 29 Stress Fatigue prior predictive plots

No significant improvement here but this can again be explained by the Spearman and Pearson plot of the independent variables to the response variable, which demonstrated that there was a very weak linear relationship between these variables, but a more moderate linear relationship for the Strain Fatigue dataset.

This is consistent with the different performance scores of each dataset even though different Regression models are used, since these Regression models assume a linear relationship to pick up on.

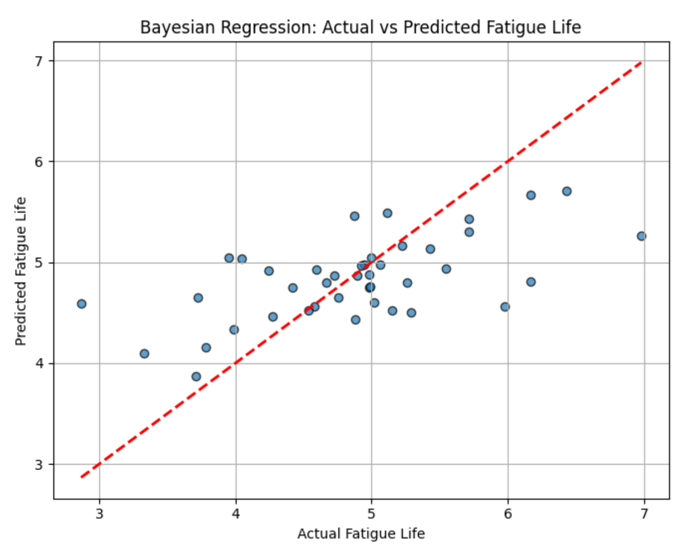


Figure 30 Actual vs Predicted Stress Fatigue life

```
Bayesian Regression Performance on Test Set:
MSE: 0.43152778571517686
MAE: 0.48077877869978164
R2 Score: 0.3573076696994528
```

Figure 31 Results of Bayesian Regression on Stress Fatigue life

The trace plots exhibit expected behavior, with some deviation from the original coefficient values. The original input coefficients for this run had the intercept mean as 4.8, intercept mean variance as 0.4, coefficient sigma as 1.0, and overall dataset variation as 0.50. However, the Posterior distribution had updated intercept mean to be around 4.7, and overall dataset variance around 0.55, indicating the data had more noise than originally anticipated.

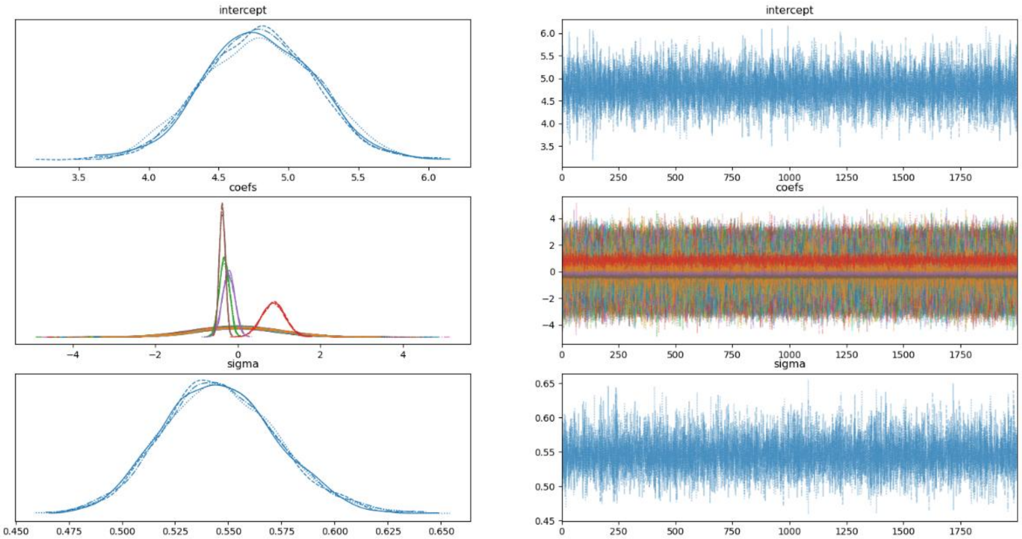


Figure 33 Trace plots for Stress Fatigue Bayesian Regression run

The large number of independent variables (approx. 400) could only mean a fraction of the coefficient plots could be displayed, however, the overall trend displays that for some coefficients, there are a large distribution/uncertainty, but for many other variables the model is confident about the mean value of coefficients. Most of the coefficients are then to be said to have a negative impact on the response variable, which is again expected.

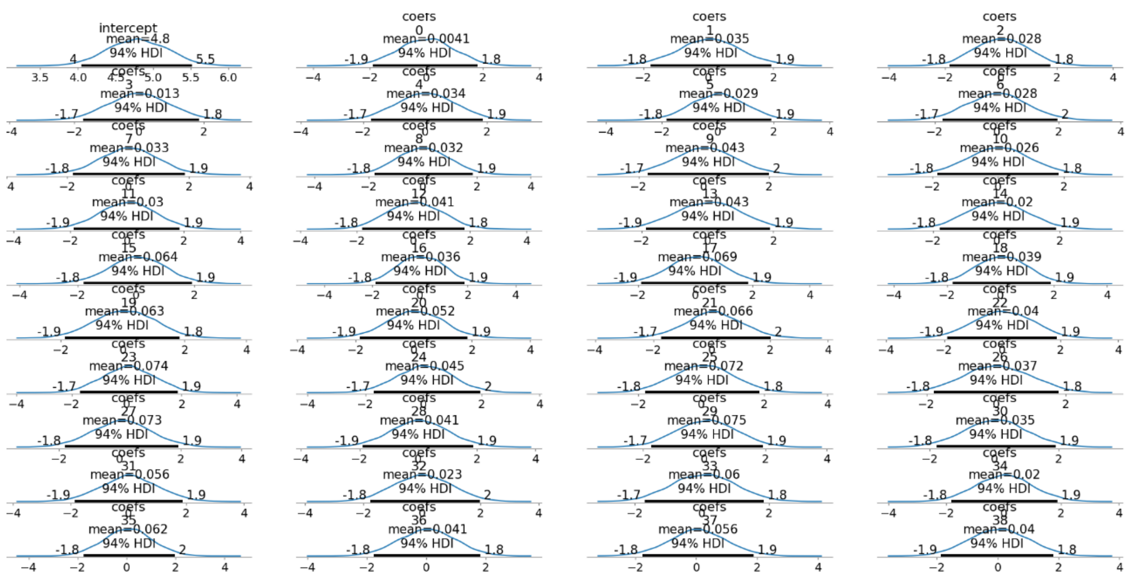


Figure 32 coefficient plots for Stress Fatigue Bayesian Regression run

	mean	sd	hdi_3%	hdi_97%	mcse_mean	mcse_sd	ess_bulk	ess_tail	r_hat
intercept	4.79	0.40	4.04	5.51	0.00	0.00	19464.81	5716.16	1.0
coefs[0]	0.00	1.00	-1.90	1.81	0.01	0.01	16865.39	5482.07	1.0
coefs[1]	0.04	0.98	-1.78	1.94	0.01	0.01	15280.86	5520.91	1.0
coefs[2]	0.03	0.97	-1.82	1.78	0.01	0.01	12538.53	5830.02	1.0
coefs[3]	0.01	0.96	-1.74	1.85	0.01	0.01	17421.38	6004.40	1.0
...	...	...	...	...	...	...	...	...	...
coefs[482]	-0.32	0.12	-0.55	-0.10	0.00	0.00	4647.37	5906.22	1.0
coefs[483]	0.86	0.28	0.34	1.38	0.00	0.00	4432.57	5386.01	1.0
coefs[484]	-0.22	0.15	-0.50	0.06	0.00	0.00	4687.22	5653.07	1.0
coefs[485]	-0.37	0.07	-0.51	-0.24	0.00	0.00	7493.15	6080.20	1.0
sigma	0.55	0.03	0.49	0.60	0.00	0.00	15070.35	5609.26	1.0

Figure 34 Stress Fatigue data convergence table

3. Beam Dataset Analysis- Gaussian Process Regression

The raw independent variable data was transformed (Standardized), kept the same as all other Regression data. The response variable transformation was log transformed.

Displacement Data

For the Displacement Dataset, five different Kernels were experimented with, with WhiteKernel utilized for all kernels, with the value as the standard deviation of the response variable. The Radial Basis Function (RBF) kernel itself was used for its ability to model smooth functions.

The physical behavior in the beam might show localized irregularities, so Matern kernels with ν=1.5 and ν=2.5 were also chosen, which allow for moderate and smoother variations. The Rational Quadratic kernel was utilized because it can be interpreted as a mixture of RBF kernels, capturing variability at multiple scales—a useful feature when both global trends and local stress concentrations are present.

Additionally, a mixed kernel combining a periodic component (ExpSineSquared) with an RBF was tested to determine if any cyclic patterns exist in the stress distribution, although periodicity was not expected to dominate the behavior. This diverse set of kernels enabled us to comprehensively assess the underlying data structure and select the model that best fits the observed stress responses.

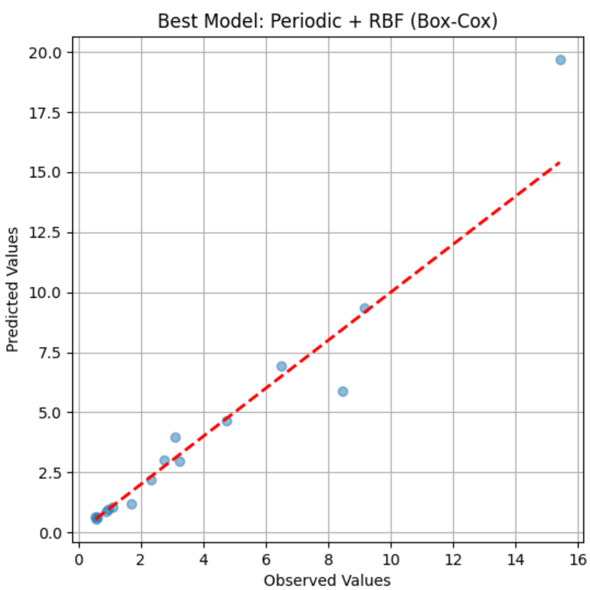
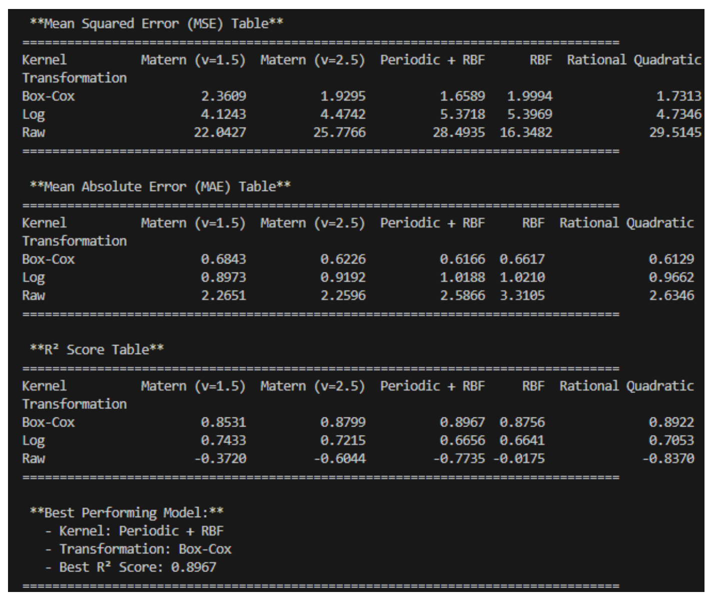


Figure 35 Observed vs Predicted values for GPR of Beam displacement data



Mean Squared Error (MSE) Table

Kernel / Transformation	Matern (v=1.5)	Matern (v=2.5)	Periodic + RBF	RBF	Rational Quadratic
Box-Cox	2.3609	1.9295	1.6589	1.9994	1.7313
Log	4.1243	4.4742	5.3718	5.3969	4.7346
Raw	22.0427	25.7766	28.4935	16.3482	29.5145

Mean Absolute Error (MAE) Table

Kernel / Transformation	Matern (v=1.5)	Matern (v=2.5)	Periodic + RBF	RBF	Rational Quadratic
Box-Cox	0.6843	0.6226	0.6166	0.6617	0.6129
Log	0.8973	0.9192	1.0188	1.0210	0.9662
Raw	2.2651	2.2596	2.5866	3.3105	2.6346

R^2 Score Table

Kernel / Transformation	Matern (v=1.5)	Matern (v=2.5)	Periodic + RBF	RBF	Rational Quadratic
Box-Cox	0.8531	0.8799	0.8967	0.8756	0.8922
Log	0.7433	0.7215	0.6656	0.6641	0.7053
Raw	-0.3720	-0.6044	-0.7735	-0.0175	-0.8370

Best Performing Model:
- Kernel: Periodic + RBF
- Transformation: Box-Cox
- Best R^2 Score: 0.8967

Figure 36 Kernel performance table comparison for Beam displacement data

The periodic+ RBF kernel mix performed the best, with the highest R squared value of all Regression runs performed so far (0.8967) for the Displacement Data. The MAE and MSE results are 0.6166 and 1.6589 respectively. These results are the best performing results yet.

One potential explanation for this, is that even though the stress isn't periodic, the periodic kernel may capture a pattern in the stress data that the RBF itself might not adequately capture. Combined, the RBF + Periodic kernel can fully observe and detail any subtle fluctuations in stress previously unseen.

Another potential explanation is that the periodic kernel can "modulate" the response from the RBF kernel. The potential variability in stress detailed by the periodic kernel can explain why the overall fit is the best. This may be due to localized, unrealistic stress concentrations caused by voids being too close to the edge of the beam.

Stress Data

The same kernel combinations discussed in the Displacement Data section above were utilized for the same reason, and the best result was chosen by finding the highest R squared value.

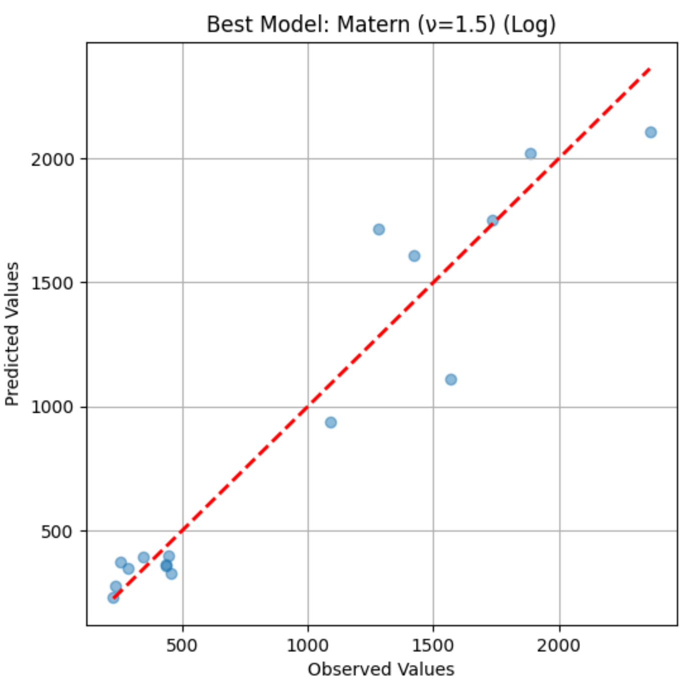


Figure 38 Observed vs Predicted values for GPR of Beam stress data

Mean Squared Error (MSE) Table

Kernel Transformation	Matern (v=1.5)	Matern (v=2.5)	Periodic + RBF	RBF	Rational Quadratic
Box-Cox	38295.3265	42723.4060	42184.2497	44783.5193	42214.1011
Log	36964.9708	41053.8689	42900.3136	42958.3523	41275.6416
Raw	46081.0918	489028.0249	44286.9641	489029.1296	44285.7831

Mean Absolute Error (MAE) Table

Kernel Transformation	Matern (v=1.5)	Matern (v=2.5)	Periodic + RBF	RBF	Rational Quadratic
Box-Cox	143.0306	152.6057	151.1759	156.8606	151.2748
Log	139.7522	148.3378	152.1916	152.2940	148.7549
Raw	180.2507	643.5420	164.7195	643.5428	164.7128

R² Score Table

Kernel Transformation	Matern (v=1.5)	Matern (v=2.5)	Periodic + RBF	RBF	Rational Quadratic
Box-Cox	0.9187	0.9092	0.9104	0.9049	0.9103
Log	0.9215	0.9128	0.9089	0.9087	0.9123
Raw	0.9021	-0.0388	0.9059	-0.0388	0.9059

Best Performing Model:
- Kernel: Matern (v=1.5)
- Transformation: Log
- Best R² Score: 0.9215

Figure 37 Kernel performance table comparison for Beam stress data

The Matern kernel with v = 1.5 performed the best, as it encourages a balance between smoothness and flexibility. The resulting R squared value is 0.9215 and the MAE and MSE scores were 139.75 and 3631.64 respectively.

This kernel creates roughness so it can capture local variability across scales as opposed to RBF which assumes infinitely smooth functions, which may help explain irregularities in the displacement data. This can help the model's flexibility to understand small and sudden changes in the data which is important for the beam dataset. However, all the kernel functions have relatively high R squared scores (around 0.90) so the performance of all the kernels here were acceptable.

Between the Stress and Displacement datasets for all the runs, it can again be noted that for the lower Stress/Displacement values, the test model outputs were clustered near the ideal line but tended to disperse more around the ideal line for upper Stress/Displacement values. This indicates more runs within this range would be ideal and may help with the model's performance.

4. External Dataset Analysis- Gaussian Process Regression

The raw independent variable data was transformed (Standardized), kept the same as all other Regression data. The response variable was not transformed, due to it being already having a normalized distribution/log transformed already.

Strain Fatigue

As discussed for the Beam dataset, the same five kernel functions were tested, and the best performing kernel was the Rational Quadratic. The R squared value for this was 0.8664, and MAE and MSE values of 0.197 and 0.0844 respectively. This is the best performance of a Regression model so far, seeing for example the R squared value for the Strain Fatigue set for Lasso and Ridge Regression was 0.7765 and 0.7635.

One possibility why Rational Quadratic kernel performed best (but keep in mind the performance of other kernels was close to this) is because it can capture long term trends and short-term fluctuations in the strain data for each metal alloy. This implies that for the Strain Fatigue data, there might be a global trend that plays a role in predicting the fatigue, but there's other localized details that are important too such as the microstructure in each alloy.

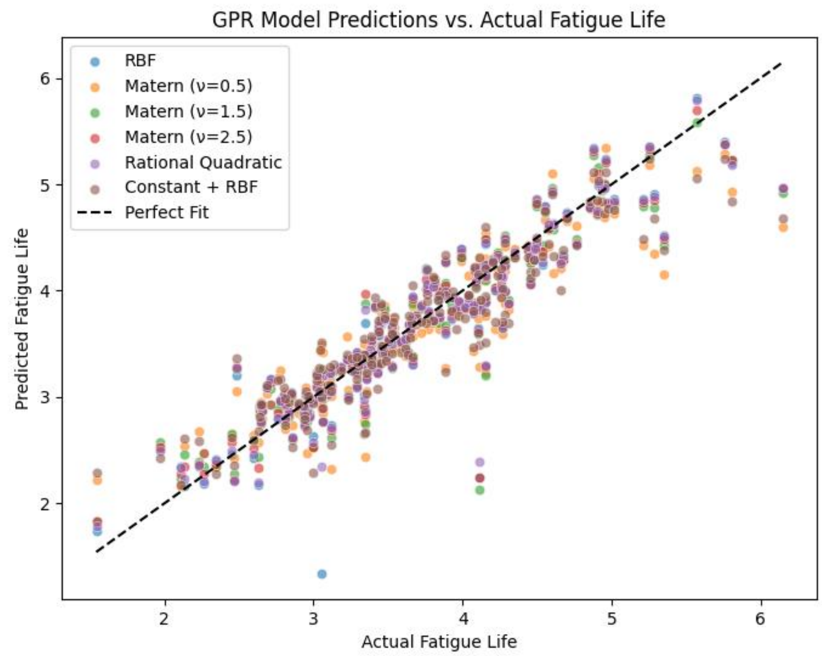


Figure 39 Actual vs Predicted Strain Fatigue life

```
 **GPR Model Performance Comparison**
=====================================================
              Kernel       MSE       MAE  R2 Score
4 Rational Quadratic  0.084473  0.197304  0.866392
3      Matern (ν=2.5)  0.084858  0.195059  0.865783
2      Matern (ν=1.5)  0.088694  0.195180  0.859716
0                 RBF  0.098789  0.200831  0.843750
5      Constant + RBF  0.104708  0.235272  0.834387
1      Matern (ν=0.5)  0.109890  0.232997  0.826190
=====================================================

 **Best Performing GPR Model:**
   - Kernel: Rational Quadratic
   - Best R² Score: 0.8664
=====================================================
```

Figure 40 Kernel performance comparison table

Stress Fatigue

Identical setup and procedure to the Strain Fatigue run, but this time, the best performing kernel mix was the Matern (v =1.5) kernel. The R squared value was 0.8751 and the MAE and MSE scores were 0.219 and 0.0838 respectively.

This is the best performance for a Regression model yet, as the Lasso/Ridge Regression models produced poor R squared values, 0.3273 and 0.3583 respectively.

The Matern kernel performed the best, which may be explained that stress would be sensitive to local effects like stress concentrations or cracks, and the Matern kernel is a reasonable choice to pick to capture these local effects of roughness or sudden changes that play a more important role than the Strain Fatigue data, as Rational Quadratic performed best for that set.

This is not unexpected since the evaluation of Spearman and Pearson Coefficients from the first assignment, showcased that there is a very weak linear relationship between the independent variables and response variable.

GPR is not a linear relationship but rather a nonparametric Bayesian method with no assumptions about the relationship. It is very useful for capturing trends in nonlinear relationships, and it is demonstrated here as it predicts the test data samples very well comparatively to the Ridge/Lasso Regression.

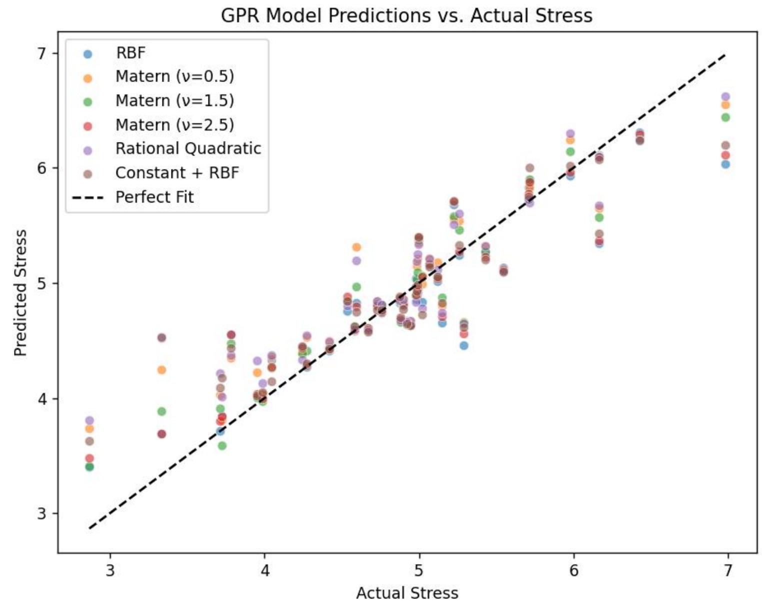


Figure 41 Actual vs Predicted Stress for Stress Fatigue data

```
**GPR Model Performance Comparison**
======================================================
               Kernel       MSE       MAE  R2 Score
2      Matern (v=1.5)  0.083833  0.219561  0.875144
3      Matern (v=2.5)  0.104895  0.219676  0.843775
0                 RBF  0.111543  0.228543  0.833875
1      Matern (v=0.5)  0.115079  0.255738  0.828607
5      Constant + RBF  0.138351  0.258365  0.793948
4  Rational Quadratic  0.138748  0.280510  0.793357
======================================================

**Best Performing GPR Model:**
  - Kernel: Matern (v=1.5)
  - Best R² Score: 0.8751
======================================================
```

Figure 42 Kernel performance table for Stress Fatigue data

Conclusion for Bayesian Linea Regression/GPR runs

Overall, for the Beam dataset, the Bayesian Linear Regression could have been improved with additional data from Abaqus, seeing that most the erroneous prediction points were at the upper stress/displacement levels. Obtaining more data for higher regions could have improved the model runs. Additional high-level data helps models understand these nonlinear effects, significantly improving prediction accuracy. Adjusting priors even more could have improve inference with the limited data though.

For the external dataset, Bayesian Linear Regression proved to struggle with the large dataset, requiring the use of PCA to reduce dimensionality and enable the model to converge. This shows that too little or too much data can prove troublesome for Bayesian Linear Regression.

For Gaussian Process Regression, the size of the dataset for both the Beam and External dataset showed no challenges in performance, seeing that this Regression run provided the best results seen for both datasets.

5. Research Paper on Bayesian Multimodal Regression

Posterior Shrinkage Larger datasets lead to posterior distributions of regression weights that are more peaked, indicating greater certainty in parameter estimates. Increased data volume allows likelihood to dominate the priors, ensuring the posterior estimates converge toward the true data-driven values. However, smaller datasets, like my beam data, produce less peaked posterior distributions when compared to external datasets, indicating higher uncertainty.

Model Uncertainty Reduction Smaller datasets have ambiguity regarding which predictors are truly influential. Limited data may leave marginal probabilities intermediate, showing uncertainty about feature relevance. With larger datasets, predictor importance emerges more clearly, producing confident inclusion probabilities. In my beam dataset, limited observations particularly at higher stress or displacement levels result in uncertain or unstable predictor selections.

Robustness to Priors As datasets grow, results become less sensitive to the prior distributions because likelihood dominates inference. Small adjustments to priors' impact regression outcomes with smaller datasets, as I observed in my beam regressions, highlighting the sensitivity to prior assumptions in limited-data scenarios.

Predictive Performance Pure fit metrics such as R squared can mislead in small datasets by overfitting complex models. Bayesian model averaging, used in the reviewed paper, addresses overfitting by penalizing model complexity. Increased dataset sizes generally enhance sample prediction accuracy by improving differentiation between important relationships and noise. In the beam study, limited data caused noticeable performance degradation, especially at higher displacement values. More data via simulation on Abaqus would significantly improve predictive confidence.

Agreement with Authors' Results The authors' findings on Bayesian multimodal regression align with theoretical Bayesian expectations: increased data reduces posterior uncertainty, diminishes prior impact, and clarifies predictor importance. I agree with these results, as my limited beam dataset clearly illustrated how posterior uncertainty remains high and predictor influence less certain due to limited data. While the authors' conclusions make sense, they could improve their tutorial by explicitly exploring empirical thresholds at which increased sample size mitigates collinearity and reduces prior sensitivity.

The report involves subjective human responses with large sample sizes and well-defined, independently measured predictors. My dataset, by contrast, involves structural responses under mechanical loading with predictors related to geometric variables and loading conditions. These input differences are important. Consequently, Bayesian multi-model regression might show clearer inclusion probabilities and tighter posterior intervals in the paper's large, diverse dataset, while in my beam dataset, fewer runs limit differentiation between correlated predictors, increase sensitivity to prior choices, and reduce model generalization. More data is crucial for improving the predictive performance and robustness in mechanical contexts.

Practically, expanding my beam dataset with additional observations at critical regions (high displacement/stress) would reduce uncertainty and improve predictive stability, as previously discussed in the assignment.