Bayesian Data Analysis - Assignment 2 (S1889112)

The primary software used in this work was R, interfacing to `JAGS` via the libraries `rjags` and `runjags`. `JAGS` uses Gibbs sampling (a form of Markov Chain Monte Carlo (MCMC)).

1a: Exploratory Data Analysis

The results of the EDA are in Figure 1. Audouin abundance has fallen over time, while Yellowlegged abundance has increased over time. As such, we may expect a negative relationship between the two types of gulls. However looking at the correlation coefficient for all the data suggests no relationship between Audouin and Yellowlegged gulls. This is possibly due to the spikes in Audouin gulls in the mid to late 90s (before a sudden drop-off), where there is also a spike in Yellowlegged gulls. The small sample (26) means these values may have quite a large effect on the correlation. Sequentially removing the 2 outliers indicated by the boxplot leads to correlations of -0.07, and -0.30 for removing Audouin abundances of 625 and 525 respectively. The red and green lines are the cases with 625 and 525 removed respectively.

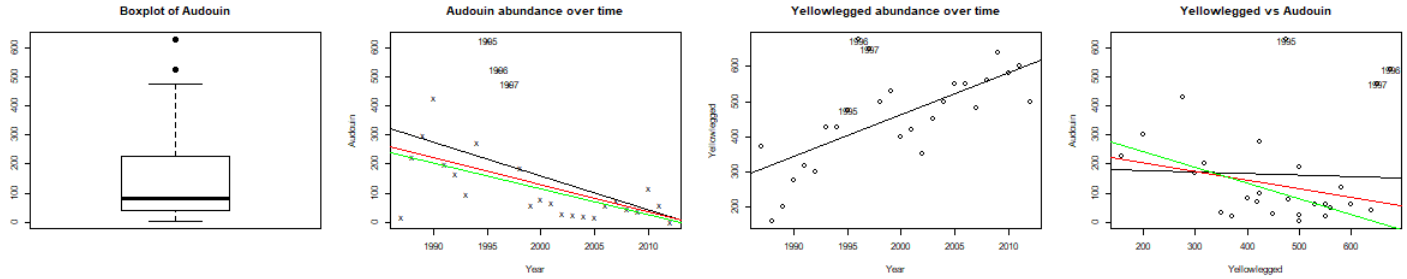


Figure 1: EDA Plots

1b: Initial Model

The only covariate is the year, which has been demeaned to assist convergence. The priors on the β_j are set to be $N(\mu_{\beta_j} = 0, \sigma^2_{\beta_j} = 100)$ to reflect a lack of information about the parameters, while avoiding the issue of sparse priors causing numerical issues. Other priors were tested, all giving similar results. This was done by building the model for precision parameters (0.001, 0.01, 0.1, 1, 10). The code for this is in the "Consistency Check" section inside the R code for this question. The model is as follows:

$$
\begin{aligned}
&\textbf{Likelihood:} && y_i|\mu_i, Year_i \sim Poisson(\mu_i), \quad i = 1, \dots n \\
&\textbf{Link:} && log(\mu_i) = \beta_0 + \beta_1(Year_i - \overline{Year}) \\
&\textbf{Priors:} && \beta_j \sim N(0, 100), \quad j = 0, 1
\end{aligned}
$$

The model was run with a burn-in of 20000, 20000 iterations, and a thinning interval of 4. Effective sample sizes are approximately 13500 for both β_0 and β_1 using the combined chains, with initial values determined by JAGS, due to the "Node inconsistent with parents" error when initialised manually or through a function call.
The Gelman-Rubin convergence diagnostic has point estimates and upper confidence intervals of 1. This is replicated by running the appended code. Trace and density plots are shown in Figure 2, showing good mixing and smoothness.

The results of the parameter estimates can be seen in Figure 3 to 3 decimal places. The 95% credible intervals for both parameters are very narrow. The value for β_0 suggests that in an average year, the log mean of Audouin abundance is 4.943, with a 95% credible interval of (4.909, 4.978). β_1 suggests that each extra year is associated with a fall in the log mean of -0.076, with a 95% credible interval of (-0.08, -0.071). In level terms this can be written $(\mu|Year = x + 1) = e^{-0.076}(\mu|Year = x)$.

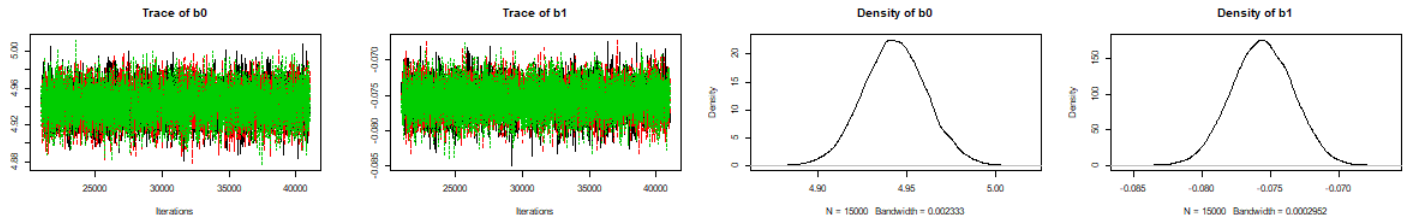


Figure 2: Trace and Density Plots

	2.5%	25%	50%	75%	97.5%	Mean	SD	Naive SE	Time-series SE	95% CI	95% CI Width
b0	4.909	4.931	4.943	4.955	4.978	4.943	0.018	0	0	(4.909,4.978)	0.069
b1	-0.08	-0.077	-0.076	-0.074	-0.071	-0.076	0.002	0	0	(-0.08,-0.071)	0.009

Figure 3: Results of Poisson Regression $log(\mu_i) = \beta_0 + \beta_1(Year_i - \overline{Year})$

1c: Extra-Variance

The objective of this section is to add variance to the model manually, so it is not solely determined by the mean, as is the case for a Poisson random variable. We augment the model from 1b to include a unit-specific error term in the link. That is:

$$\begin{aligned}
&\textbf{Likelihood:} && y_i|\mu_i, Year_i \sim Poisson(\mu_i), \quad i = 1, \dots n \\
&\textbf{Link:} && log(\mu_i) = \beta_0 + \beta_1(Year_i - \overline{Year}) + \epsilon_i \\
&\textbf{Priors:} && \epsilon_i \sim N(0, \sigma_\epsilon^2), \quad \beta_j \sim N(0, 100), \quad \forall i, j \\
&\textbf{Hyperprior:} && \sigma_\epsilon \sim U(0, 10)
\end{aligned}$$

There is now much higher autocorrelation in the parameters. Hence we use a burn-in of 100000, 1 million iterations, and a thinning interval of 400. 3 chains are used, with initial values determined by JAGS. The thinning interval was determined using a 2-step process. Firstly, running a chain with no thinning, and seeing which parameter has the lowest effective sample size. Secondly, examining the effective sample size for this parameter for various thinning intervals by manually thinning the chain and computing the effective size on this thinned chain. This is shown in the `thincheck` function. We did not check multiple alternative priors, due to computational infeasibility. However changing the standard deviation of the β_j to 1 yielded no significant differences.
The effective sample sizes are at least 4000. The Gelman-Rubin diagnostic is 1 for point estimates and upper confidence intervals, and the trace plots (Figure 5) suggest good mixing. The density plots are again smooth for all parameters.
The results can be seen in Figure 4. The credible intervals are now much wider, and the values for both β_j have shifted downwards. The extra-variance term in the linear component has approximately unit standard deviation, with a reasonably narrow 95% credible interval of (0.79, 1.44). Note that exponentiating this error, along with the multiplicative impact, suggests a reasonably large error with regards to the mean of raw series. β_0 has a mean of 4.5, with a 95% credible interval of (4.07, 4.92), while β_1 has a mean of -0.09 with a 95% credible interval of (-0.15, -0.04). Interpretation is the same as previously.

	2.5%	25%	50%	75%	97.5%	Mean	SD	Naive SE	Time-series SE	95% CI	95% CI Width
b0	4.071	4.365	4.502	4.645	4.921	4.503	0.215	0.002	0.003	(4.071,4.921)	0.85
b1	-0.149	-0.112	-0.093	-0.074	-0.037	-0.093	0.029	0	0	(-0.149,-0.037)	0.112
sigma.epsilon	0.781	0.936	1.036	1.154	1.459	1.057	0.171	0.002	0.002	(0.781,1.459)	0.678

Figure 4: Results of Poisson Regression $log(\mu_i) = \beta_0 + \beta_1(Year_i - \overline{Year}) + \epsilon_i$

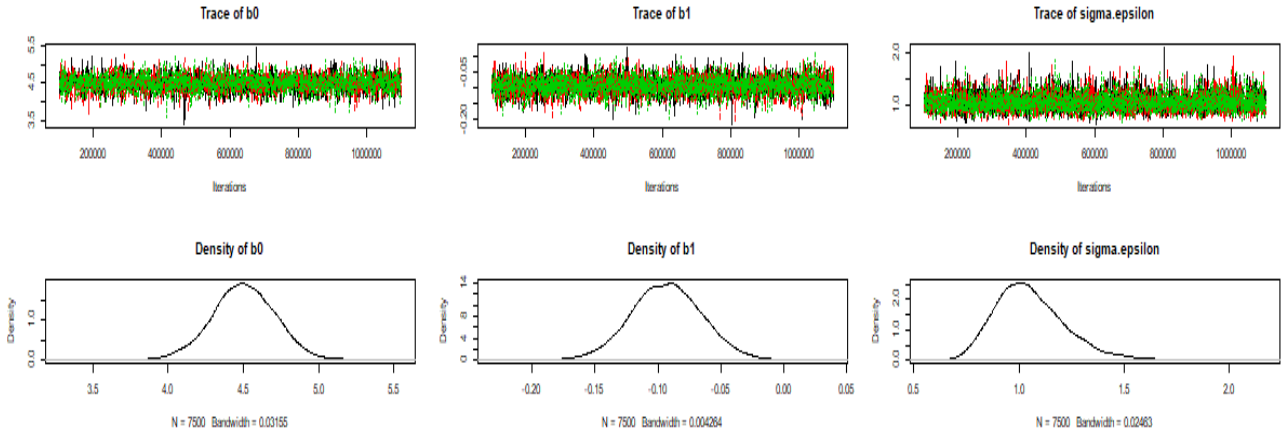


Figure 5: Trace and Density Plots

1d: Yellowlegged as a Covariate

Adding the abundance of Yellowlegged gulls is just a change in the link to include the covariate. Hence we have the same model as in 1c (with the same prior on β_2 as the other β_j) but with the following link:

$$\textbf{Link:} \qquad log(\mu_i) = \beta_0 + \beta_1(Year_i - \overline{Year}) + \beta_2(Yellowlegged_i - \overline{Yellowlegged}) + \epsilon_i$$

Using the same thinning evaluation as the previous section, we run 1.4 million iterations, thinning of 500, a burn-in of 200000, and 3 chains initialised by JAGS. Trace plots are in Figure 6, with full results in Figure 7. The trace plots show good mixing and the density plots are smooth. Effective sample sizes are at least 2500, with Gelman-Rubin diagnostics of 1 for all parameters.

The posterior parameter on β_2 has a mean of 0.004, and a 95% credible interval of (0, 0.008). This appears to be economically insignificant, in that the value of the coefficient is essentially 0. Based on the exploratory analysis this might be expected, given that the complete-case analysis (shown by the black line) suggests no relationship. However once we remove the 2 large outliers, there is a visible negative relationship. This would fit with intuition, since an increase in Yellowlegged gulls means food supplies are reduced once Audouin gulls arrived (since they have a later reproduction cycle). Given that all cases were used, we expected an insignificant relationship between the number of Yellowlegged gulls and the number of Audouin gulls in the same year, but should be wary that this lack of a relationship could be due to some environmental shock in 1996 and 1997 causing the abundance of both gulls to spike.

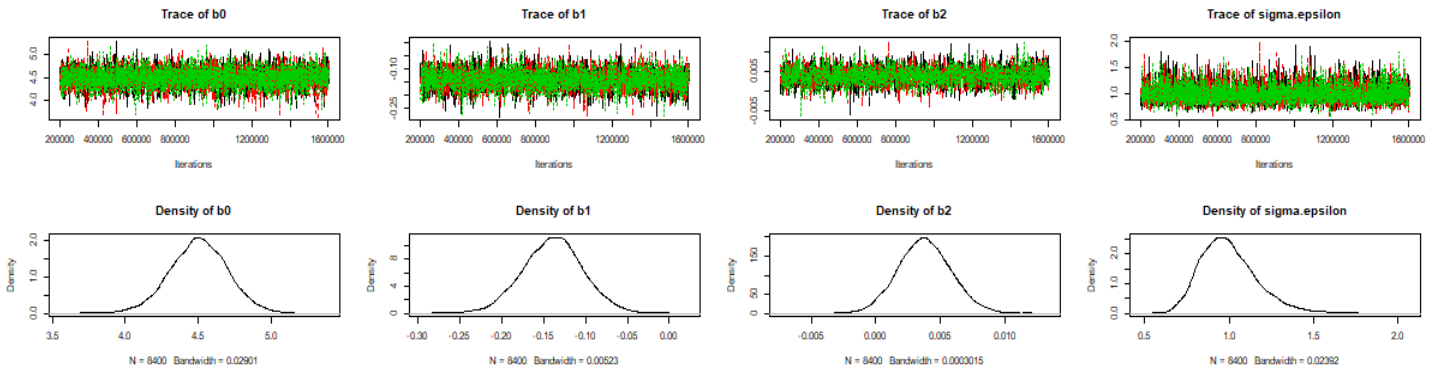


Figure 6: Trace and Density Plots

	2.5%	25%	50%	75%	97.5%	Mean	SD	Naive SE	Time-series SE	95% CI	95% CI Width
b0	4.104	4.373	4.505	4.636	4.892	4.503	0.199	0.002	0.003	(4.104,4.892)	0.788
b1	-0.211	-0.163	-0.139	-0.115	-0.067	-0.139	0.037	0	0.001	(-0.211,-0.067)	0.144
b2	0	0.002	0.004	0.005	0.008	0.004	0.002	0	0	(0,0.008)	0.008
sigma.epsilon	0.724	0.88	0.98	1.097	1.393	1	0.17	0.002	0.002	(0.724,1.393)	0.669

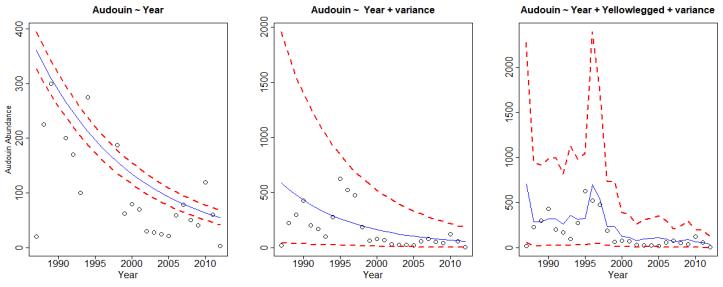
Figure 7: Results of Poisson Regression $log(\mu_i) = \beta_0 + \beta_1(Year_i - \overline{Year}) + \beta_2(Yellowlegged_i - \overline{Yellowlegged}) + \epsilon_i$

1e: Posterior Predictive

Suppose we have S MCMC samples. For each observation i in the data, we compute $\mu_i = exp(\beta_0 + \beta_1(Year_i - \overline{Year}) + \beta_2(Yellowlegged_i - \overline{Yellowlegged}) + \epsilon_i$ repeatedly for each set of posterior samples of the β_j and ϵ. Therefore each observation has S predictions based on the different samples. To sample ϵ, we form a vector based on the samples of σ_ϵ. For each sample of σ_ϵ, we randomly draw from a normal distribution with mean 0 and standard deviation equal to the sample from σ_ϵ, and store the result in the vector. Then we can denote the s^{th} sample of the i^{th} observation as:

$$\mu_{is} = exp(\beta_{0s} + \beta_{1s}(Year_i - \overline{Year}) + \beta_{2s}(Yellowlegged_i - \overline{Yellowlegged}) + \epsilon_s)$$

Where 's' is the index of the MCMC sample. Define the vector of posterior predictive samples for observation i as $\boldsymbol{\mu_i} = (\mu_{i1} \quad \mu_{i2} \ldots \mu_{iS})$. The mean can be calculated for each observation by taking the mean of this vector, and the 90% credible interval by passing the vector through the `quantile` function with `probs = c(0.05, 0.95)`. For each μ_{is} we sample from the Poisson distribution with parameter μ_{is}, creating the vector $\boldsymbol{y_i} = (y_{i1} \ldots y_{iS})$ for each observation. That is $y_{is} \sim Pois(\mu_{is})$. The results of this analysis are shown in Figure 8. The dashed red lines represent the limits of the credible interval, and the blue line is the mean.



	Mean 1b	Lower 1b	Upper 1b	Mean 1c	Lower 1c	Upper 1c	Mean 1d	Lower 1d	Upper 1d
1987	361	327	395	590	44	1961	707	59	2274
1988	335	302	368	531	40	1748	289	21	948
1989	310	279	341	479	37	1545	286	23	917
1990	288	258	318	433	34	1404	318	28	990
1991	267	239	295	392	31	1269	321	29	1003
1992	247	221	275	354	28	1132	261	24	816
1993	229	204	255	321	26	1030	361	33	1127
1994	212	188	238	290	24	924	312	29	980
1995	197	174	221	264	22	848	329	31	1046
1996	183	160	205	239	20	760	702	49	2396
1997	169	148	192	217	18	684	532	42	1761
1998	157	136	178	198	17	631	236	22	741
1999	146	126	166	180	15	577	231	21	733
2000	135	116	155	164	14	521	122	11	391
2001	125	107	145	150	12	477	115	11	364
2002	116	98	135	137	11	438	80	6	258
2003	108	90	125	125	10	397	97	9	310
2004	100	83	117	114	9	361	102	9	324
2005	92	77	109	104	8	332	108	10	350
2006	86	70	102	96	7	308	94	8	302
2007	80	65	95	88	7	279	64	5	208
2008	74	60	89	80	6	255	75	6	243
2009	68	55	83	74	5	239	89	7	294
2010	63	50	77	68	5	219	62	5	204
2011	59	46	73	62	4	199	59	4	194
2012	55	42	68	58	4	188	36	2	120

a: Posterior Predictive Mean and 90% Credible Interval **b:** Full Results for Posterior Predictive

Figure 8: Posterior Predictive Graph and Full Results for all models

1f: Model Selection

Given the results in Figure 9, we can be reasonably confident that the model from 1b is not appropriate to model the data, caused by the lack of an extra-variation term leading to the 90% credible intervals capturing the observed data incredibly poorly since the Poisson distribution has variance equal to the mean. Just including this extra-variation allows the credible interval to capture the data well in terms of the credible interval. However the mean is quite poorly captured, with large residuals until about the year 2000. In terms of the mean, the data is captured better in model 1d, however this could be due to overfitting, since we have only 9 (26/3) observations per β_j.

It appears as though most of the relationship can be captured by a time trend, coinciding with the exploratory analysis done in 1a, and the small value of the coefficient on the demeaned Yellowlegged covariate in 1d.

Both of the models with the extra-variance term seem adequate to model the data, given the near-identical DIC, and the plots capturing the data. If we are looking to establish some semblance of causality, I would select model 1d, as the presence of Yellowlegged gulls dramatically improves how well the mean captures the data, despite the potential overfitting, and is intuitively an important variable to control for. Additionally, despite

mentioning the possibility of overfitting, the DIC is in fact the lowest among the models suggesting the improvement in predictive performance may be worth the loss in parsimony.

	Model 1B	Model 1C	Model 1D
DIC	2986.13	216.6	216.54

Figure 9: DIC Comparison

2a: EDA

The results of basic EDA are shown in Figure 10. The top left panel shows the correlation between the environment variables of the farm. In calculating this, we only took 1 row from each farm to avoid inflated correlations caused by an unequal number of observations for each farm. For instance suppose we have 2 farms with 100 observations each, and a 3rd farm with 1 observation. Correlation between any 2 variables will be almost perfect, as the first 2 farms dominate the sample. We see that temperature, height, and rain are all intricately related, with high correlations between each of the 3 covariates. Conversely, the slope and permeability have almost no linear relationship with the other covariates.

The other panels in this row calculate the parasite proportion within each farm, and for each cow age. These suggest that farm 1 is 'safest', given its large sample size. The farms with higher parasite proportions have smaller sample sizes, but farms 6, 7, and 13 appear to be the most parasitic farms. There appears to be a higher parasite probability in older cows.

The remaining panels show the distribution of each covariate given a specific parasite level. In addition, the Mann-Whitney p-value is reported. This is a nonparametric test for differences in the median of 2 groups, and does not lose much power from a t-test. Given this statistic, it appears as if there is clear discrepancy in the medians between groups for temperature, rainfall, and height. We conclude from this simple test that there is not much evidence for median difference between the groups for slope, age, and permeability. However we note that permeability only takes on 2 values, so a test for median differences should not be relied upon to draw conclusions. For instance, (0, 0, 0, 0, 0, 1, 1, 1, 1, 1) and (0, 0, 0, 0, 0, 0, 0, 0, 0, 1) will have the same median despite the groups clearly being different.

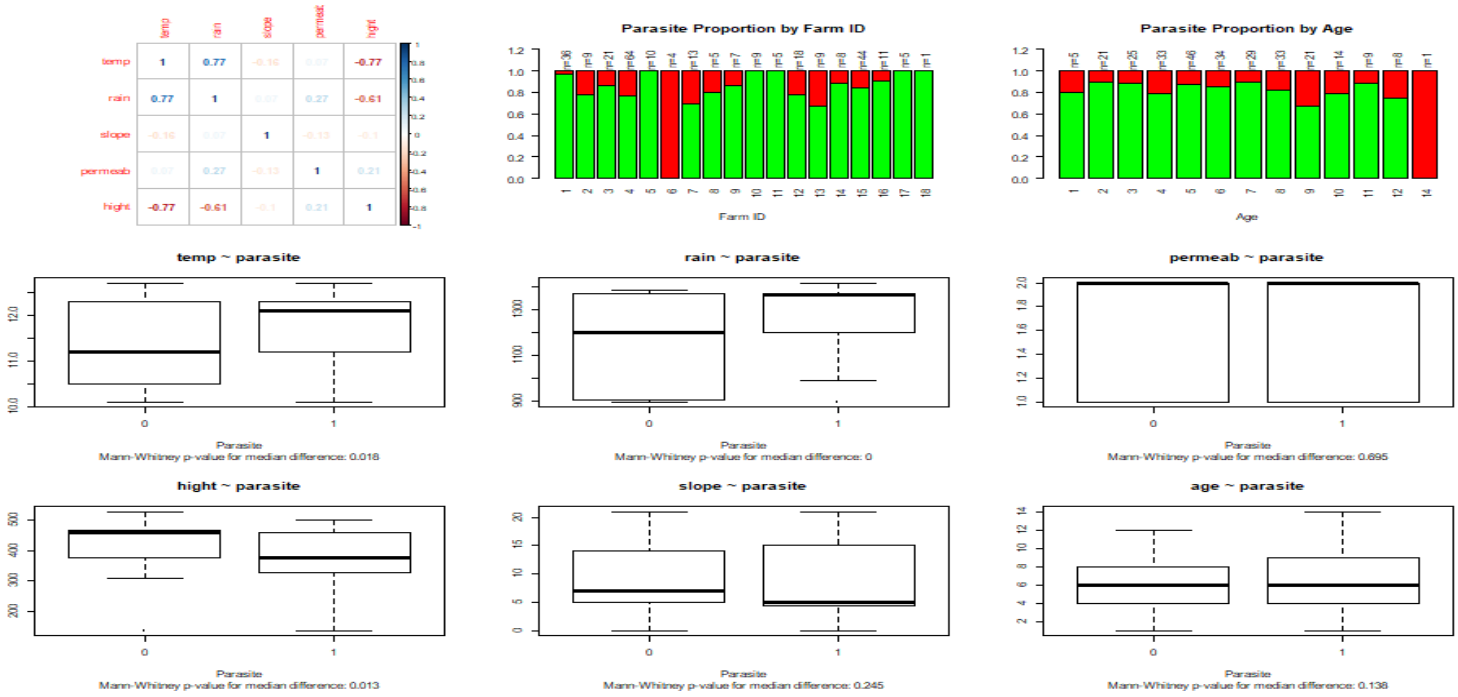


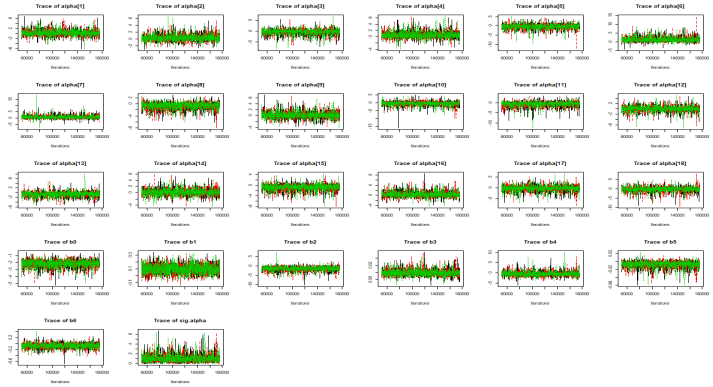
Figure 10: EDA Plots

2b: Model, All covariates

All covariates have been demeaned to aid convergence. The demeaned version of covariate x_k will be represented as $\widehat{x}_k$. The vector of demeaned variables is $(\widehat{x}_1, \widehat{x}_2, \widehat{x}_3, \widehat{x}_4, \widehat{x}_5, \widehat{x}_6)$, corresponding respectively to age, temperature, rain, permeability, height, and slope. We represent the random effect as the sum of the common mean β_0 and a Gaussian white noise term α_j. The sum of these can be seen as the total random effect. In general, let us represent the total random intercept as $\gamma_j \sim N(\mu_\gamma, \sigma^2_\gamma)$, this is equivalent to $\mu_\gamma + \epsilon_j$, where $\epsilon_j \sim N(0, \sigma^2_\gamma)$, by properties of the normal distribution. In both cases, we place priors on μ_γ and σ^2_γ, representing the common distribution of the random intercept. Referring back to our model, we have $\mu_\gamma = \beta_0$ and $\epsilon_j = \alpha_j$. β_0 has the same prior as the other β_k, where $\beta_k \sim N(0, 100)$, however this will be referred to as a hyperprior to make the relation to the random effect more clear. Instead of placing a prior on the variance, we place it on the standard deviation, such that $\sigma_\alpha \sim U(0, 10)$.

Likelihood:	$y_{ij}\|p_{ij}, x_i \sim Bernoulli(p_{ij}) \qquad \forall i, j$
Link:	$Logit(p_{ij}) = \beta_0 + \alpha_j + \beta_1\widehat{x}_{1,ij} + \beta_2\widehat{x}_{2,ij} + \beta_3\widehat{x}_{3,ij} + \beta_4\widehat{x}_{4,ij} + \beta_5\widehat{x}_{5,ij} + \beta_6\widehat{x}_{6,ij}$
Priors:	$\alpha_j \sim N(0, \sigma^2_\alpha), \quad \beta_k \sim N(0, 100), \quad i \in (1, \ldots, n), j \in (1, \ldots, 18), k \in (1, \ldots, 6)$
Hyperpriors:	$\sigma_\alpha \sim U(0, 10) \quad \beta_0 \sim N(0, 100)$

The thinning interval was determined in the same manner as specified in 1c. A burn-in of 50000 was used, with 125000 iterations and a thinning interval of 100. Effective sample sizes were at least 1800 for all parameters. Trace plots and results are in Figure 11. Trace plots show good mixing, and the highest Gelman-Rubin upper confidence interval was 1.05. Initial values were determined by JAGS, due to the same error as previously. For all future models, initial values will also be determined by JAGS. β_0 has a mean value of -2.144, with a 95% credible interval of (-3.2, -1.4), suggesting there is quite a lot of uncertainty around the common mean of the random effect. In general, there is a high degree of uncertainty around almost all of the parameters, with the width of the credible intervals being around the range (3,5) for all parameters except the overall mean (β_0), and β_1, β_3, β_5 and β_6. These are the coefficients on age, rain, height, and slope respectively. Of these, β_1 (age) seems to be the most economically significant, with a mean of 0.099, but its credible interval (-0.03, 0.23) contains 0 casting doubt on the significance of age. However there is a 93% probability this parameter is greater than 0, suggesting it is likely to be positive (see R code). Of the parameters with narrow credible intervals, only β_5 has a credible interval not containing 0, suggesting it is the most statistically significant. As such, of the covariates, it seems age (economic significance) and height (statistical significant) have the strongest relationship with the parasite probability.



a: Trace Plots

	2.5%	25%	50%	75%	97.5%	Mean	SD	Naive SE	Time-series SE	95% CI	95% CI Width
alpha[1]	-1.768	-0.342	0.064	0.621	2.566	0.176	1.051	0.017	0.017	(-1.768,2.566)	4.334
alpha[2]	-1.146	-0.152	0.209	0.79	2.33	0.352	0.872	0.014	0.016	(-1.146,2.33)	3.476
alpha[3]	-3.245	-0.849	-0.215	0.201	1.563	-0.4	1.173	0.019	0.023	(-3.245,1.563)	4.808
alpha[4]	-1.208	-0.047	0.354	0.974	2.938	0.514	1.003	0.016	0.023	(-1.208,2.938)	4.146
alpha[5]	-3.114	-0.765	-0.14	0.278	1.978	-0.282	1.206	0.02	0.02	(-3.114,1.978)	5.092
alpha[6]	-0.651	0.214	0.828	1.693	4.425	1.104	1.31	0.021	0.023	(-0.651,4.425)	5.076
alpha[7]	-1.011	0.022	0.505	1.294	3.626	0.76	1.224	0.02	0.022	(-1.011,3.626)	4.637
alpha[8]	-3.152	-1.018	-0.34	0.065	1.078	-0.551	1.045	0.017	0.02	(-3.152,1.078)	4.23
alpha[9]	-1.742	-0.321	0.084	0.629	2.37	0.173	0.993	0.016	0.017	(-1.742,2.37)	4.112
alpha[10]	-3.277	-0.816	-0.167	0.227	1.666	-0.361	1.225	0.02	0.022	(-3.277,1.666)	4.943
alpha[11]	-3.384	-1.023	-0.34	0.074	1.204	-0.568	1.163	0.019	0.019	(-3.384,1.204)	4.588
alpha[12]	-2.072	-0.355	0.04	0.508	1.963	0.041	0.967	0.016	0.017	(-2.072,1.963)	4.035
alpha[13]	-2.898	-1.058	-0.413	0.003	1.168	-0.56	1.005	0.016	0.018	(-2.898,1.168)	4.066
alpha[14]	-1.692	-0.316	0.079	0.646	2.234	0.167	0.946	0.015	0.016	(-1.692,2.234)	3.926
alpha[15]	-3.371	-1.203	-0.525	-0.027	1.02	-0.703	1.103	0.018	0.02	(-3.371,1.02)	4.391
alpha[16]	-1.335	-0.169	0.224	0.86	2.882	0.391	1.03	0.017	0.019	(-1.335,2.882)	4.217
alpha[17]	-2.994	-0.664	-0.047	0.429	2.157	-0.159	1.225	0.02	0.02	(-2.994,2.157)	5.151
alpha[18]	-3.047	-0.706	-0.082	0.369	2.048	-0.214	1.217	0.02	0.019	(-3.047,2.048)	5.095
b0	-3.243	-2.367	-2.091	-1.847	-1.336	-2.142	0.474	0.008	0.009	(-3.243,-1.336)	1.907
b1	-0.028	0.054	0.098	0.141	0.229	0.098	0.065	0.001	0.001	(-0.028,0.229)	0.257
b2	-4.139	-1.995	-1.257	-0.66	0.519	-1.392	1.183	0.019	0.026	(-4.139,0.519)	4.658
b3	0.003	0.007	0.009	0.012	0.02	0.01	0.004	0	0	(0.003,0.02)	0.017
b4	-2.347	-1.143	-0.597	0.081	2.002	-0.478	1.119	0.018	0.022	(-2.347,2.002)	4.349
b5	-0.028	-0.011	-0.006	-0.002	0.004	-0.007	0.008	0	0	(-0.028,0.004)	0.032
b6	-0.244	-0.115	-0.071	-0.029	0.078	-0.074	0.079	0.001	0.001	(-0.244,0.078)	0.322
sig.alpha	0.088	0.568	0.948	1.411	2.997	1.085	0.764	0.012	0.017	(0.088,2.997)	2.909

b: Full Results

Figure 11: Results and Convergence for model 2b

2c: Model simplification

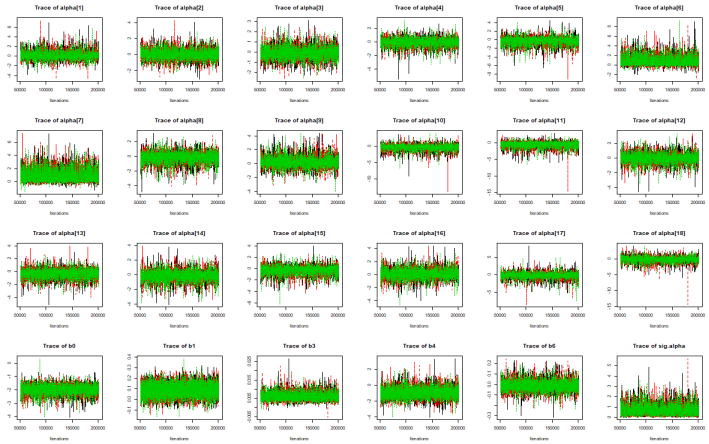
Now we look at the correlations between the environment variables to determine some model simplifications, and therefore refer back to Figure 10, looking at the top left correlation plot. We can see that temperature is highly correlated with both rain and height, and therefore remove it, as we might suggest that we can explain most of the influence of temperature by these 2 variables. Hence we try a model without temperature. We then see that rain and height are highly correlated with each other, and therefore remove both variables individually, as one may capture the effect of the other. As such we try 3 simplifications. Firstly, a model without temperature. Secondly, a model without temperature or height. Thirdly, a model without temperature or rain. The coefficient indices will be maintained for consistency. E.g. the coefficient on permeability remains β_4 throughout. We rely on DIC to select the model. For computational speed, each model was run with only 2 chains for the DIC calculation. A burn-in of 50000 was used, with 100000 iterations for each DIC sample. The results of can be seen in Figure 12. We have assumed convergence after a burn-in of 10000, and that there is a reasonable effective sample size. This is perhaps not unreasonable, given the effective sizes for previous and future models, but is by no means a guarantee. This was not done due to time constraints. We select the model which has removed temperature and height as covariates, as it has the lowest DIC.

	Full Model	Temp Removed	Temp, Rain Removed	Temp, Height Removed
DIC	242.65	243.45	244.04	242.24

Figure 12: DIC of considered models

We recompile the model with 3 chains, a burn-in of 50000, 150000 iterations, and a thinning interval of 50. The effective sizes are at least 4000. The Gelman-Rubin diagnostic upper intervals are below 1.02, and the trace plots in Figure 13 show good mixing.

In this same Figure, we see the width of credible intervals for the parameters has generally fallen. The mean for β_1 (age) is mostly unchanged, but the mean for β_4 (permeability) has increased in magnitude (decreased in value) from -0.467 to -0.917, suggesting some of the predictive power lost by dropping β_2 and β_5 has been captured. The coefficient on rain β_3 is largely unchanged, falling from 0.01 to 0.006. This could be due to the omitted variables pulling the coefficient in different directions as a result of the opposite sign correlations (rain is positively correlated with temperature, negatively with height). The coefficient on the slope, β_6, shrinks in importance, falling in magnitude from -0.075 to -0.014. The parameters on many of the random effects have significantly changed. For example farms 2 to 4 now have farm-specific effects much closer to 0.



a: Trace Plots

	2.5%	25%	50%	75%	97.5%	Mean	SD	Naive SE	Time-series SE	95% CI	95% CI Width
alpha[1]	-1.261	-0.204	0.033	0.401	1.956	0.133	0.758	0.008	0.008	(-1.261,1.956)	3.217
alpha[2]	-1.12	-0.227	0.012	0.31	1.27	0.043	0.564	0.006	0.006	(-1.12,1.27)	2.39
alpha[3]	-1.175	-0.345	-0.051	0.152	1.03	-0.086	0.518	0.005	0.006	(-1.175,1.03)	2.205
alpha[4]	-1.212	-0.27	-0.007	0.211	1.026	-0.039	0.538	0.006	0.006	(-1.212,1.026)	2.238
alpha[5]	-2.152	-0.515	-0.077	0.139	1.149	-0.223	0.796	0.008	0.009	(-2.152,1.149)	3.301
alpha[6]	-0.405	0.045	0.438	1.151	3.21	0.717	0.96	0.01	0.013	(-0.405,3.21)	3.615
alpha[7]	-0.241	0.11	0.6	1.35	3.118	0.841	0.924	0.01	0.013	(-0.241,3.118)	3.359
alpha[8]	-1.602	-0.358	-0.021	0.209	1.106	-0.097	0.645	0.007	0.007	(-1.602,1.106)	2.708
alpha[9]	-1.199	-0.166	0.061	0.451	1.806	0.157	0.692	0.007	0.008	(-1.199,1.806)	3.005
alpha[10]	-2.16	-0.523	-0.088	0.119	1.129	-0.244	0.802	0.008	0.009	(-2.16,1.129)	3.289
alpha[11]	-2.438	-0.74	-0.197	0.04	0.73	-0.419	0.819	0.009	0.01	(-2.438,0.73)	3.168
alpha[12]	-1.335	-0.254	0	0.267	1.179	-0.008	0.591	0.006	0.007	(-1.335,1.179)	2.514
alpha[13]	-1.86	-0.577	-0.143	0.064	0.855	-0.276	0.659	0.007	0.008	(-1.86,0.855)	2.715
alpha[14]	-1.501	-0.364	-0.031	0.205	1.17	-0.083	0.635	0.007	0.007	(-1.501,1.17)	2.671
alpha[15]	-1.812	-0.522	-0.111	0.098	0.972	-0.229	0.669	0.007	0.008	(-1.812,0.972)	2.784
alpha[16]	-1.318	-0.217	0.028	0.37	1.542	0.075	0.67	0.007	0.007	(-1.318,1.542)	2.86
alpha[17]	-1.977	-0.434	-0.037	0.204	1.355	-0.135	0.796	0.008	0.008	(-1.977,1.355)	3.332
alpha[18]	-2.054	-0.4	-0.026	0.224	1.444	-0.118	0.825	0.009	0.009	(-2.054,1.444)	3.498
b0	-2.757	-2.171	-1.954	-1.764	-1.374	-1.981	0.342	0.004	0.004	(-2.757,-1.374)	1.383
b1	-0.036	0.045	0.088	0.131	0.213	0.088	0.064	0.001	0.001	(-0.036,0.213)	0.249
b3	0.003	0.005	0.006	0.007	0.011	0.006	0.002	0	0	(0.003,0.011)	0.008
b4	-2.177	-1.311	-0.919	-0.537	0.413	-0.916	0.646	0.007	0.007	(-2.177,0.413)	2.59
b6	-0.106	-0.04	-0.014	0.013	0.088	-0.013	0.048	0.001	0.001	(-0.106,0.088)	0.194
sig.alpha	0.028	0.286	0.59	0.954	1.964	0.683	0.523	0.006	0.009	(0.028,1.964)	1.936

b: Full Results

Figure 13: Results and Convergence for no temperature, no height model

2d: Posterior for farms 1 and 6

The calculation method for the posterior is very similar to that used in 1e, but we are reporting on the link component, and not sampling from the distribution of the outcome variable. That is we are reporting p (the probability of a cow from farm j having a parasite), not y (the classification of a cow). We know the environmental characteristics of a specific farm are fixed, and assume the age to be equal to the mean for the following calculations. We calculate the posterior probability for farms 1 and 6 for each of the S MCMC samples we have, in the following manner:

$$logit(p_{js}) = \beta_{0s} + \alpha_{js} + \beta_{1s}\widehat{x}_{1j} + \beta_{3s}\widehat{x}_{3j} + \beta_{4s}\widehat{x}_{4j} + \beta_{6s}\widehat{x}_{6j}$$

Where the $\widehat{x}$ are defined as in 2b, s denotes the MCMC sample index, and the subscript i is dropped since we assume age equal to the mean, hence the 3rd term on the right hand side is zero by definition of $\widehat{x}$, which removes all cow individuality in the model. After we calculate the linear combination, we apply `inverse.logit` from the `boot` package to calculate the probabilities. We then have a vector $\boldsymbol{p_j}$ with S probabilities corresponding to each MCMC sample. The mean is calculated by taking the average of this vector. The 95% symmetric credible interval is calculated by using the `quantile` function with argument `probs = c(0.025, 0.975)`, and the probability of an epidemic is equivalent to the proportion of elements of the relevant $\boldsymbol{p_j}$ greater than 0.2.

A histogram for each farm is illustrated, as well as the exact figures. These are in Figures 15 and 14. The mean is marked by a blue vertical line, and the 95% credible interval by red lines. Note that figures are rounded to 3 decimal places, and we used a limited number of samples. Hence an epidemic is not 'impossible' for farm 1, but is very unlikely. This makes sense given the fairly large sample for farm 1 (n= 36 as per 2a), with only 1 parasitic cow. The expected proportion of cows on farm 1 (with an average age) with a parasite is 2.5%, and there is a 95% probability this proportion is between 0.2% and 7.6%.
Similar logic applies to farm 6. All 4 cows on the farm have a parasite, which would be very unlikely with a probability of below 20%. With a probability of exactly 20%, all 4 cows having a parasite has a probability of 0.16% assuming independence (`dbinom(4, 4, 0.2)` in R). The low sample size leads to a wide credible interval of (0.223, 0.963), suggesting there is a 95% probability that the proportion of cows on farm 6 with the parasite (with an average age) is between 22.3% and 96.1%.

	Mean	95% CI	P(Epidemic)
Farm 1 (Rand)	0.025	(0.002, 0.074)	0
Farm 6 (Rand)	0.593	(0.226, 0.961)	0.989

Figure 14: Posterior Parasite Probabilities (Random Effects)

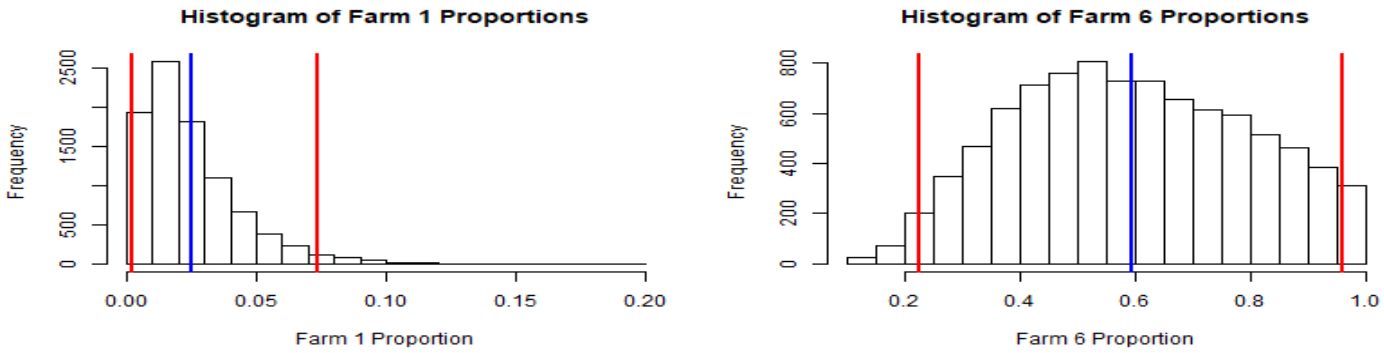


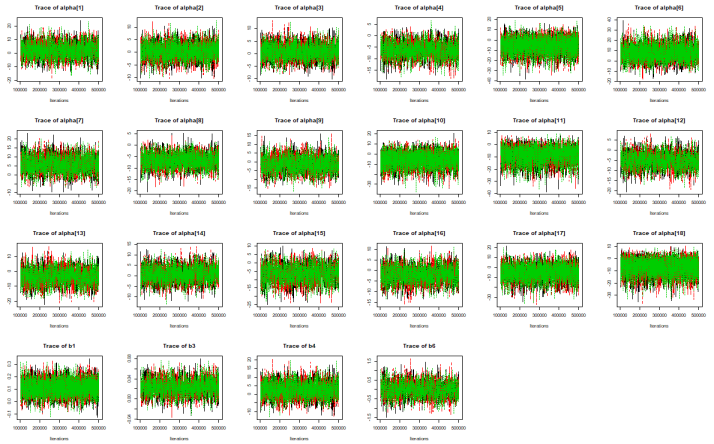
Figure 15: Histogram of Posterior Parasite Probabilities (Random Effects)

Q2e: Fixed Effects

In estimating a fixed effects model, each farm-specific intercept has its own hyperparameters, meaning the model has the same likelihood, but the following link and priors:

$$\begin{aligned}
&\textbf{Likelihood:} && y_{ij}|p_{ij}, x_i \sim Bernoulli(p_{ij}) \qquad \forall i, j \\
&\textbf{Link:} && Logit(p_{ij}) = \alpha_j + \beta_1 \widehat{x}_{1,ij} + \beta_3 \widehat{x}_{3,ij} + \beta_4 \widehat{x}_{4,ij} + \beta_6 \widehat{x}_{6,ij} \\
&\textbf{Priors:} && \alpha_j \sim N(0, 100), \quad \beta_k \sim N(0, 100), \quad \forall j, k
\end{aligned}$$

The model was run with burn-in 100000, 400000 iterations, thinning of 200, and 3 chains. Effective sizes were at least 1500, and Gelman-Rubin diagnostics were 1 for all upper intervals. The relevant trace plots are in Figure 16, along with the full results. The trace plots show good mixing. The posterior summaries are in Figure 17.
The credible intervals for the parameters are now much wider, since there is no information sharing via the common distribution, with 95% credible interval widths of between 10 and 30 for the farm-specific intercepts.
The expected value of the proportion of parasitic cows has increased for both farms. From 2.5% to 2.9% for farm 1, and drastically from 59% to 98% for farm 6. The reason for farm 6 shooting up is likely because of the lack of information sharing between the groups, as they no longer come from a common distribution. To 3 decimal places, there is almost total certainty about the probability of an epidemic with fixed effects (of these farms). The credible interval is now slightly wider for farm 1, and much narrower for farm 6.



a: Trace Plots

	2.5%	25%	50%	75%	97.5%	Mean	SD	Naive SE	Time-series SE	95% CI	95% CI Width
alpha[1]	-8.5	-1.502	2.139	5.704	12.782	2.151	5.392	0.07	0.113	(-8.5,12.782)	21.282
alpha[2]	-5.687	-1.756	0.363	2.455	6.443	0.362	3.106	0.04	0.058	(-5.687,6.443)	12.13
alpha[3]	-5.079	-1.259	0.697	2.789	6.848	0.76	3.048	0.039	0.061	(-5.079,6.848)	11.927
alpha[4]	-13.04	-8.441	-6.035	-3.781	0.506	-6.131	3.428	0.044	0.081	(-13.04,0.506)	13.546
alpha[5]	-21.932	-9.947	-4.41	0.522	8.526	-5.012	7.79	0.101	0.106	(-21.932,8.526)	30.458
alpha[6]	-6.782	2.33	7.253	12.418	23.463	7.494	7.627	0.098	0.127	(-6.782,23.463)	30.245
alpha[7]	-2.972	2.748	5.941	9.092	15.011	5.957	4.608	0.059	0.104	(-2.972,15.011)	17.983
alpha[8]	-12.826	-8.872	-6.719	-4.554	-0.617	-6.723	3.141	0.041	0.063	(-12.826,-0.617)	12.209
alpha[9]	-9.915	-4.216	-1.324	1.826	7.506	-1.223	4.421	0.057	0.106	(-9.915,7.506)	17.421
alpha[10]	-21.462	-9.933	-4.31	0.504	8.414	-4.969	7.666	0.099	0.103	(-21.462,8.414)	29.876
alpha[11]	-22.608	-11.983	-7.262	-3.249	3.066	-7.953	6.611	0.085	0.094	(-22.608,3.066)	25.674
alpha[12]	-12.763	-7.786	-5.087	-2.576	2.2	-5.166	3.834	0.049	0.095	(-12.763,2.2)	14.963
alpha[13]	-13.962	-7.058	-3.374	0.338	7.104	-3.386	5.427	0.07	0.127	(-13.962,7.104)	21.066
alpha[14]	-6.283	-1.653	0.662	3.21	7.855	0.767	3.617	0.047	0.071	(-6.283,7.855)	14.138
alpha[15]	-17.164	-10.197	-6.53	-2.945	3.425	-6.577	5.284	0.068	0.131	(-17.164,3.425)	20.589
alpha[16]	-9.611	-4.599	-1.945	0.651	5.442	-2.006	3.842	0.05	0.096	(-9.611,5.442)	15.053
alpha[17]	-21.096	-9.728	-4.307	0.71	9.324	-4.712	7.828	0.101	0.108	(-21.096,9.324)	30.42
alpha[18]	-22.131	-10.778	-5.736	-1.67	4.488	-6.59	6.817	0.088	0.093	(-22.131,4.488)	26.619
b1	-0.019	0.066	0.11	0.153	0.238	0.109	0.065	0.001	0.001	(-0.019,0.238)	0.257
b3	-0.007	0.013	0.023	0.034	0.055	0.024	0.016	0	0	(-0.007,0.055)	0.062
b4	-6.044	-0.382	2.68	5.677	11.561	2.695	4.498	0.058	0.087	(-6.044,11.561)	17.605
b6	-0.798	-0.272	-0.004	0.282	0.825	0.005	0.413	0.005	0.011	(-0.798,0.825)	1.623

b: Full Results

Figure 16: Results and Convergence for no temperature, no height model (Fixed Effects)

	Mean	95% CI	P(Epidemic)
Farm 1 (Fixed)	0.02892	(0.00072, 0.1025)	0.00017
Farm 6 (Fixed)	0.97996	(0.74967, 1)	1

Figure 17: Posterior Parasite Probabilities (Fixed Effects)

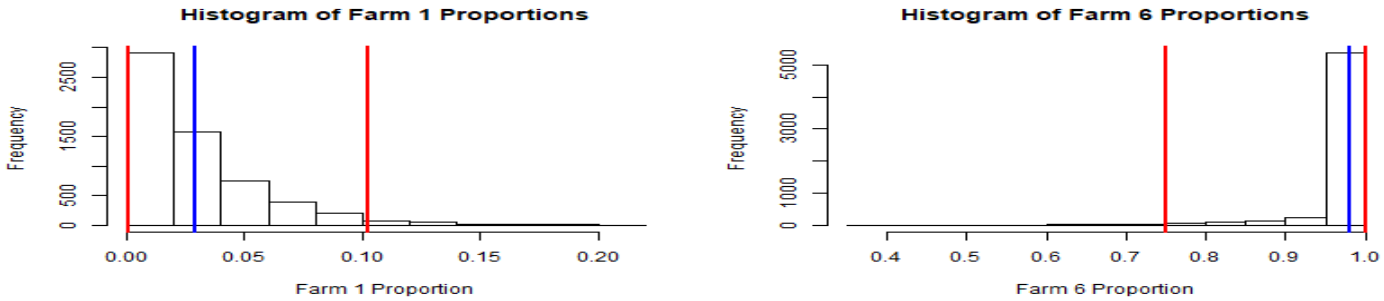


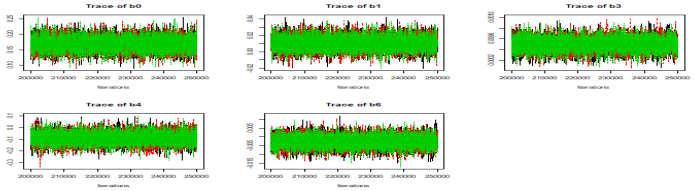
Figure 18: Posterior Parasite Probabilities (Fixed Effects)

Q3: A New Model

This section explores the performance of a Bayesian linear probability model (BLPM) applied to the cows data. We will use the reduced form of the model from 2c, and use a fixed intercept. That is, we are using a very simple model to examine a baseline for the model. Results will be compared using DIC. Additionally, a normal likelihood will be used. We place a conjugate, pseudo-uninformative gamma prior on $\tau = \frac{1}{\sigma^2}$.

$$y_i|\mu_i, x_i, \sigma^2 \sim N(\mu_i, \sigma^2) \qquad \forall i$$
$$\mu_i = \beta_0 + \beta_1\widehat{x}_{1i} + \beta_3\widehat{x}_{3i} + \beta_4\widehat{x}_{4i} + \beta_6\widehat{x}_{6i}$$
$$\tau \sim \Gamma(0.01, 0.01) \qquad \beta_j \sim N(0, 0.01) \quad j \in \{0, 1, 3, 4, 6\}$$

Here, μ_i represents the probability observation i has a parasite. To calculate the DIC, 3 chains were used with a burn-in of 50000, and 150000 iterations. The results were then sampled using 50000 further iterations, with thinning of 10 for computational ease. The DIC was 236.6, which is lower than all of the models under the hierarchical GLM framework (Figure 12. There is no reason to suspect a lack of convergence based on the trace plots in Figure 19, while the Gelman-Rubin statistics were 1 across the board. Effective sample sizes were all around 15000.



	2.5%	25%	50%	75%	97.5%	Mean	SD	Naive SE	Time-series SE	95% CI	95% CI Width
b0	0.1262	0.1539	0.1683	0.183	0.2117	0.1685	0.0219	0.0002	0.0002	(0.1262,0.2117)	0.0855
b1	-0.0054	0.0049	0.0104	0.0159	0.0269	0.0105	0.0082	0.0001	0.0001	(-0.0054,0.0269)	0.0323
b3	0.0003	0.0004	0.0005	0.0006	0.0007	0.0005	0.0001	0	0	(0.0003,0.0007)	0.0004
b4	-0.1859	-0.1232	-0.0902	-0.0563	0.0101	-0.0896	0.0502	0.0004	0.0004	(-0.1859,0.0101)	0.196
b6	-0.0095	-0.0051	-0.0028	-0.0005	0.0038	-0.0028	0.0034	0	0	(-0.0095,0.0038)	0.0133

a: Trace Plots **b:** Full Results

Figure 19: Results and Convergence for no temperature, no height model (BLPM)

With the BLPM, β_0 represents the proportion of cows with a parasite. The mean is 16.8% (the proportion in the dataset), with a 95% credible interval of (12.5%, 21.2%). β_3, the coefficient on rainfall is 0.0005, suggesting that ceteris paribus, an increase in rainfall of 100ml is associated with a 0.05 (5%) increase in the probability a cow has a parasite. For this parameter, all samples were positive suggesting a very high probability that, when controlling for the variables in the model, there is a positive association between rainfall and the probability a cow has a parasite. This positive association is consistent with the results from Figure 13. The probability each parameter has the same sign as its mean is (1, 0.898, 1, 0.9654, 0.8) for $(\beta_0, \beta_1, \beta_3, \beta_4, \beta_5)$. These same probabilities from the model in 2c are (0.923, 0.999, 0.927, 0.65) for β_1 to β_6 (β_0 not included due to the random intercept distorting these probabilities). There appears to be consistency here, with β_6 having the most uncertain sign consistency across both models, while the positive effect of β_3 remains near certain.
Intuitively it seems problematic that this model has a lower DIC, since the output 'probability' can leave the (0,1) bound, and there is no distinction between the farms. The reason for this is due to the unbalanced data, in that the number of cows without parasites far outweighs those with parasites. As such, by setting all parameters to 0 (except the intercept), it is still possible to attain an 83% classification accuracy, given a standard cutoff of 0.5 as everything will be classified as 0 (no parasite). This is what the BLPM is doing (classifying everything as 0). See the `PREDICTIONS` section in the R code for details. As such, it appears that the hierarchical random effects model is hardly outperforming this basic model in terms of classification, with a higher penalty due to complexity, leading to higher DIC. As such the random effects model seems reasonably poor given this basic metric, and we need more data to form a reasonable model for the parasite probabilities, since at the moment (in terms of classification) even just a linear intercept model would hardly lose any predictive power vs the random effects model.

A Appendix: Functions

```
#### DEMEAN ####

demean <- function(x) {x - mean(x)}



#### DENSITY PLOTTER ####


mcmc.dens <- function(combined.results) {
  #
  # combined results is of class "mcmc".
  # Output you would get from mcmc.combine(coda.samples(...))
  #

  # Dataframe conversion
  mcmc.df <- as.data.frame(combined.results)

  # Parameter names
  param.names <- colnames(mcmc.df)

  for (name in param.names){
    # Parameter vector
    param.samples <- mcmc.df[, name]
    main = paste("Density of", name)
    plot(density(param.samples), main = main)
  }
}


#### THINNING CHECKER ####

thincheck <- function(results.obj, var.idx, thinhigh, thinby, chainsize){
  # var.idx: Specify variable to observe ESS for
  # chainsize: Size of each individual chain in results.obj
  #
  #
  #
  #
  #

  # Thinning intervals to consider, slice to remove 0
  thins <- seq(from = 0, to = thinhigh, by = thinby)
  thins <- thins[2:length(thins)]

  # Vector to store effective sample sizes corresponding to
  # various thinning intervals
  ess   <- numeric(length(thins))

  # i: Index of vector add to
```

```
  i     <- 1

  for (thin in thins) {
    # Create indices to take from posterior sample
    thin.vec <- seq(from=1, to=chainsize, by = thin)
    # Get thinned sample by slicing posterior sample using
    # above indices
    tempsamp <- results.obj[[1]][, var.idx][thin.vec]
    # Multiply by 3 to get ESS of combined chain
    ess[i]   <- 3*effectiveSize(tempsamp)
    i = i + 1
  }
  plot(thins, ess)
}


#### RESULTS TABLE ####

results.table <- function(combres, dig = 3){
  sum.res <- summary(combres)
  quants.res <- sum.res$quantiles
  stats.res  <- sum.res$statistics
  rep.res    <- cbind(quants.res, stats.res)
  rep.res    <- round(rep.res, dig)

  CI.95      <- paste('(', rep.res[, '2.5%'], ',', rep.res[, '97.5%'], ')', sep='')
  CI.95.width <- round(rep.res[, '97.5%'] - rep.res[, '2.5%'], dig)

  rep.res <- cbind(rep.res, CI.95)
  colnames(rep.res) <- replace(colnames(rep.res), length(colnames(rep.res)), '95% CI')

  rep.res <- cbind(rep.res, CI.95.width)
  colnames(rep.res) <- replace(colnames(rep.res), length(colnames(rep.res)), '95% CI Width')
  return(rep.res)
}


##### QUESTION 2 EXCLUSIVE #####

#### BARPLOTS FOR EDA ####

barplot.2 <- function(height.name, xlab, col = c('Green', 'Red'), srt = 90){
  ### Create barplot table
  t.height    <- table(cows$parasite, cows[, height.name])
  t.height.pc <- prop.table(t.height, margin = 2)
  t.height.pc <- round(t.height.pc, 2)
  # Get sample size to put at top of barplot
  sample.size <- apply(t.height, MARGIN = 2, sum)

  ### Create barplot ###

  main <- paste('Parasite Proportion by', Hmisc::capitalize(xlab))

```

```r
  mybar <- barplot(t.height.pc, main = main,
                   xlab = xlab, col = col,
                   ylim = c(0, 1.2), las = 2)

  ### Label with sample size
  text(mybar, y = 1.1, label = paste('n=', sample.size, sep = ''), srt = srt)

}

#### Mann-Whitney U-statistic with boxplot ####

boxplot.mw <- function(df, y.name, var.names){
  # y must be categorical, with 0 and 1

  ## Split dataset into 2 for readability
  df.0 <- df[df[, y.name] == 0, ]
  df.1 <- df[df[, y.name] == 1, ]

  for (i in 1:length(var.names)) {

    var.name <- var.names[i]
    ## Different vectors
    zero.vec <- df.0[, var.name]
    one.vec  <- df.1[, var.name]
    ## Mann-Whitney
    MW.p <- wilcox.test(zero.vec, one.vec)$p.value
    MW.p <- round(MW.p, 3)

    ## Boxplot
    main    <- paste(var.name, '~', y.name)
    sub     <- paste('Mann-Whitney p-value for median difference:', MW.p)
    xlab    <- Hmisc::capitalize(y.name)
    formula <- as.formula(main)
    boxplot(formula = formula, data = df, xlab = xlab, main = main, sub = sub)
  }
}




#### FUNCTIONS FOR QUESTION 2D AND E ####

### EXTRACT POSTERIOR SAMPLES ###

farmprobs <- function(xt, Bt, alpha){
  # xt: row of data of class matrix: Dimensions 1 x k (k parameters)
  # Bt: Matrix of coefficient samples: Dimensions k x S (S MCMC samples)
  # alpha: row of samples of farm-specific intercept, class matrix: Dimensions 1 x S

  # Add 1 on for random intercept influence to Xt, and add alpha to Bt.
  xt.a <- cbind(xt, 1)
```

```
  Bt.a <- rbind(Bt, alpha)

  ## Linear combination
  lc.farm <- xt.a%*%Bt.a

  ## Probability, plogis is the inverse logit function
  pr.farm <- plogis(lc.farm)

  ## Return probability vector
  return(pr.farm)
}


### SUMMARISE POSTERIOR SAMPLES ###

results.post  <- function(farmID, probvec, digits=2) {
  ### Get mean, CI, probability of epidemic
  mean.farm   <- mean(probvec)
  ci.farm     <- as.vector(quantile(probvec, probs = c(0.02,0.975)))
  pr.epi.farm <- mean(probvec > 0.2)

  # Get confidence interval as a string
  ci.farm.string <- paste('(', round(ci.farm[1], digits=digits), ', ',
                          round(ci.farm[2], digits=digits), ')', sep='')

  # Concatenate all results
  res <- round(cbind(ci.farm[1], ci.farm[2], pr.epi.farm, mean.farm), digits=digits)
  res <- cbind(res, ci.farm.string)
  colnames(res) <- c('2.5%', '97.5%', 'P(Epidemic)', 'Mean', '95% CI')
  rownames(res) <- paste('Farm', farmID)
  return(res)
}
```

B Appendix: Question 1

B.1 1a

```
#### Load packages and data

require(rjags)
require(runjags)
require(gridExtra)


gulls <- read.csv('Q1/Data/gulls_data.csv')


year.raw <- gulls$year

png('Q1/Q1a.png', width = 1200, height = 250)
par(mfrow = c(1, 4))

## Univariate plot of Audouin
boxplot(gulls$audouin, main = 'Boxplot of Audouin', pch = 16, cex = 1.5)


## Plot Audouin gulls per year

plot(gulls$year, gulls$audouin, xlab = 'Year', ylab = 'Audouin',
     main = 'Audouin abundance over time', pch = 'x')

abline(lm(audouin~year, data = gulls))
abline(lm(audouin~year, data = gulls[gulls$audouin < 525, ]), col = 'red')
abline(lm(audouin~year, data = gulls[gulls$audouin < 475, ]), col = 'green')


# Label years 1995 to 1997 for both time series plots
# Done by slicing the data to only include these years for the text addition

years <- c(1995, 1996, 1997)

text(x      = gulls$year[gulls$year%in%years],
     y      = gulls$audouin[gulls$year%in%years],
     labels = year.raw[gulls$year%in%years])



## Plot yellowlegged gulls against year

plot(gulls$year, gulls$yellowlegged, xlab = 'Year', ylab = 'Yellowlegged',
     main = 'Yellowlegged abundance over time')

abline(lm(yellowlegged~year, data = gulls))


```

```r
text(x      = gulls$year[gulls$year%in%years],
     y      = gulls$yellowlegged[gulls$year%in%years],
     labels = year.raw[gulls$year%in%years])


## Plot yellowlegged against Audouin

plot(gulls$yellowlegged, gulls$audouin, xlab = 'Yellowlegged', ylab = 'Audouin',
     main = 'Yellowlegged vs Audouin')
abline(lm(audouin~yellowlegged, data = gulls))
abline(lm(audouin~yellowlegged, data = gulls[gulls$audouin < 525, ]), col='red')
abline(lm(audouin~yellowlegged, data = gulls[gulls$audouin < 475, ]), col='green')

text(x      = gulls$yellowlegged[gulls$year%in%years],
     y      = gulls$audouin[gulls$year%in%years],
     labels = year.raw[gulls$year%in%years])



par(mfrow = c(1,1))
dev.off()

#### Get correlations

## All data

cor(gulls$audouin, gulls$year)
cor(gulls$audouin, gulls$yellowlegged)

#### Remove outliers for audouin

sort(gulls$audouin)

# [1]   3  20  21  25  28  30  41  50  59  60  62  70  79
# [14]  80 100 120 170 187 201 225 275 300 430 476 525 625

slice <- function(boundary) {gulls$audouin < boundary}


upp <- 525
cor(gulls$audouin[slice(upp)], gulls$yellowlegged[slice(upp)])
cor(gulls$audouin[slice(upp)], gulls$year[slice(upp)])

upp <- 625
cor(gulls$audouin[slice(upp)], gulls$yellowlegged[slice(upp)])
cor(gulls$audouin[slice(upp)], gulls$year[slice(upp)])
```

B.2 1b

```
## Demean covariates
gulls$yellowlegged <- demean(gulls$yellowlegged)
gulls$year         <- demean(gulls$year)

#### DATA ####
## Dataset
n    <- nrow(gulls)
aud  <- gulls$audouin
year <- gulls$year

## Prior
b0.mu  <- 0
b1.mu  <- 0
b0.tau <- 0.01
b1.tau <- 0.01

## data list

data  <- list(n = n, aud = aud, year = year,
              b0.mu = b0.mu, b0.tau = b0.tau,
              b1.mu = b1.mu, b1.tau = b1.tau)


#### MODEL ####

modstr.1b <- "model{

b0 ~ dnorm(b0.mu, b0.tau)
b1 ~ dnorm(b1.mu, b1.tau)

# likelihood


for (i in 1:n){
    aud[i] ~ dpois(mu[i])
    log(mu[i]) = b0 + b1*(year[i])
  }

}"


m1.b <- jags.model(textConnection(modstr.1b), data = data, n.chains = 3)

update(m1.b, 20000)


#### RESULTS ####
res.1b <- coda.samples(m1.b, c('b0', 'b1'), n.iter = 20000, thin = 4) ## thin to make following
    steps quicker

## COMBINE CHAINS
combres.1b <- combine.mcmc(res.1b)
```

```r

#### CONVERGENCE, ESS ####

png('Q1/Q1bConvergence.png', width = 1200, height = 200)

par(mfrow = c(1, 4))

traceplot(res.1b)
mcmc.dens(combres.1b)
par(mfrow=c(1,1))

dev.off()

autocorr.plot(combres.1b)
effectiveSize(combres.1b)
gelman.diag(res.1b)
gelman.plot(res.1b)

#### CONSISTENCY CHECK ####

## Change priors of b0 and b1, and then compare quantiles
## Reduced iterations for speed

modstr.1b <- modstr.1b

taus <- c(0.001, 0.01, 0.1, 1, 10)

sumlist <- list()

# Loop over index
for (tau.idx in 1:length(taus)) {

  # Extract the prior being tested
  tau.prior   <- taus[tau.idx]
  # Create label for the results to add to sumlist. of the form 'tau=0.01'
  listlabel   <- paste('tau', '=', tau.prior, sep='')

  # Alter data to have new priors, and create model
  data$b0.tau <- tau.prior
  data$b1.tau <- tau.prior
  m1.b.check  <- jags.model(textConnection(modstr.1b),
                            data = data,
                            n.chains = 1)
  # Update and extract results, create summary object from results
  update(m1.b.check, 20000)

  res.1b.check <- coda.samples(m1.b.check, c('b0', 'b1'), n.iter = 30000, thin = 4)
  sum.check    <- summary(res.1b.check)

  # Store results in sumlist
  sumlist[[listlabel]]       <- sum.check$quantiles
}
```

```
105
106 # View results
107 sumlist
108
109 #### REPORT RESULTS ####
110
111 ## EXTRACT RESULTS AS SUMMARY ##
112
113 restab.1b <- results.table(combres.1b)
114
115 png('Q1/Q1bResults.png', width = 720, height = 70)
116 grid.table(restab.1b)
117 dev.off()
```

B.3 1c

```
#### DATA RESPECIFICATION ####

data$b0.tau <- 0.01
data$b1.tau <- 0.01

#### MODEL ####

modstr.1c <- "model{


### Priors on beta
b0 ~ dnorm(b0.mu, b0.tau)
b1 ~ dnorm(b1.mu, b1.tau)

### Hyperprior information
sigma.epsilon ~ dunif(0, 10)
tau.epsilon = pow(sigma.epsilon, -2)



# likelihood
for (i in 1:n){
    aud[i] ~ dpois(mu[i])
    log(mu[i]) = b0 + b1*(year[i]) + epsilon[i]
    epsilon[i] ~ dnorm(0, tau.epsilon)
  }

}"

m1.c <- jags.model(textConnection(modstr.1c), data = data, n.chains = 3)

update(m1.c, 100000)


## 5:20 mins for 1000000 iterations, 3 chains
# Thin for ease of following steps
start_time <- Sys.time()
res.1c <- coda.samples(m1.c,
                       c('b0', 'b1', 'sigma.epsilon'),
                       n.iter = 1000000,
                       thin = 400)

end_time <- Sys.time()
end_time - start_time


# Combine results to 1 chain
combres.1c <- combine.mcmc(res.1c)


#### CHECK HOW MUCH TO THIN ####
# only run if this is 1
```

```
53
54 runthincheck <- 0
55
56 if (runthincheck == 1){
57
58 thincheck(res.1c, 1, 1000, 50, dim(res.1c[[1]])[1])
59 abline(v = c(100, 200, 300, 400, 500), col = c('red', 'blue', 'green', 'purple', 'pink'))
60
61 }
62
63
64
65
66 #### CONVERGENCE, ESS ####
67
68 ### TRACE, DENSITY PLOTS ###
69
70 png('Q1/Q1cConvergence.png', width = 1200, height = 270)
71 # 2 x 3 plot window
72 # 2: Trace and density
73 # 3: Parameters
74 par(mfrow = c(2, 3))
75
76 # All the trace plots
77 traceplot(res.1c)
78
79 # Density plot of each parameter
80 # Check functions for documentation
81 mcmc.dens(combres.1c)
82 par(mfrow=c(1,1))
83
84 dev.off()
85
86 ### AUTOCORR, GELMAN, ESS ###
87 autocorr.plot(combres.1c)
88 effectiveSize(combres.1c)
89 gelman.diag(res.1c)
90 gelman.plot(res.1c)
91
92
93 #### REPORT RESULTS ####
94
95 ## EXTRACT RESULTS AS SUMMARY ##
96 sum.1c <- summary(combres.1c)
97
98
99 ## REPORT TABLE ##
100
101 restab.1c <- results.table(combres.1c)
102
103 png('Q1/Q1cResults.png', width = 750, height=85)
104 grid.table(restab.1c)
105 dev.off()
```

B.4 1d

```
#### DATA RESPECIFICATION ####

# Already been demeaned
yellowlegged <- gulls$yellowlegged

# Add to data list
# Add after year, so next to covariates
#data <- append(data, list(yel = yellowlegged), 3)

data$yel <- yellowlegged
# Add priors to list
data$b2.mu  <- 0
data$b2.tau <- 0.01


#### MODEL ####

modstr.1d <- "model{

# Parameter priors

b0 ~ dnorm(b0.mu, b0.tau)
b1 ~ dnorm(b1.mu, b1.tau)
b2 ~ dnorm(b2.mu, b2.tau)

# Extra variation hyperprior details
# sigma uniformly distributed
# tau is inverse sigma squared (inverse variance)

sigma.epsilon ~ dunif(0, 10)
tau.epsilon <- pow(sigma.epsilon, -2)


# likelihood
for (i in 1:n){
  aud[i] ~ dpois(mu[i])
  log(mu[i]) = b0 + b1*(year[i]) + b2*(yel[i]) + epsilon[i]
    epsilon[i] ~ dnorm(0, tau.epsilon)
  }

}"


m1.d <- jags.model(textConnection(modstr.1d), data = data, n.chains = 3)

update(m1.d, 200000)


## RUNTIME: 2000000: 13m
## RUNTIME: 1500000: 10m
## RUNTIME: 1000000: 7m
start_time <- Sys.time()
```

```
res.1d <- coda.samples(m1.d, c('b0', 'b1', 'b2', 'sigma.epsilon'),
                       n.iter = 1400000, thin = 500)

end_time <- Sys.time()
end_time - start_time

effectiveSize(res.1d)

### Combined results
combres.1d <- combine.mcmc(res.1d)

#### CHECK HOW MUCH TO THIN ####
runthincheck <- 0

if (runthincheck == 1){

  thincheck(res.1d, 3, 1000, 50, dim(res.1d[[1]])[1])
  abline(v = c(100, 200, 300, 400, 500), col = c('red', 'blue', 'green', 'purple', 'pink'))

}




#### CONVERGENCE ####
## Save trace and densityplots
png('Q1/Q1dConvergence.png', width = 1200, height = 300)
par(mfrow = c(2, 4))
traceplot(res.1d)
mcmc.dens(combres.1d)
par(mfrow=c(1,1))
dev.off()

### AUTOCORR, GELMAN, ESS ###
autocorr.plot(combres.1d)
effectiveSize(combres.1d)
gelman.diag(res.1d)
gelman.plot(res.1d)



## REPORT TABLE ##

restab.1d <- results.table(combres.1d)

png('Q1/Q1dResults.png', width = 750, height=110)
grid.table(restab.1d)
dev.off()
```

B.5 1e

```r
sampler.1e.2 <- function(X, B, eps){
  # X: nxp matrix
  # B: pxS matrix
  # eps: 1xS matrix, each row same

  S  <- ncol(B)
  X  <- cbind(X, 1)
  Be <- rbind(B, eps)

  ## Linear combination and sampling
  lc   <- X%*%Be
  mu   <- exp(lc)
  samp <- t(apply(mu, 1, FUN=rpois, n = S))


  ## Summary statistics
  means <- apply(samp, 1, mean)
  lower <- apply(samp, 1, quantile, probs = 0.05)
  upper <- apply(samp, 1, quantile, probs = 0.95)

  ## Results reporting
  ret.mat <- cbind(means, lower, upper)
  colnames(ret.mat) <- c('mean', 'lower', 'upper')
  ret.df <- as.data.frame(ret.mat)


  return(ret.df)
}

#### PLOTTER.1E ####
plotter.1e <- function(ret.df, cols=c('blue', 'red', 'red'), main = '', cex = 1){

  upperlim <- max(ret.df$upper)
  ## BASE PLOT
  plot(x = year.raw, y = aud, ylim = c(0, upperlim),
       main = main, xlab = 'Year', ylab = '',
       cex = cex, cex.main = cex, cex.axis = cex, cex.lab = cex)

  ## MEAN PLOT
  lines(x = year.raw, y =ret.df$mean,   col = cols[1], pch = 'x', cex = cex)
  lines(x = year.raw, y =ret.df$lower,  col = cols[2], lwd = 2, lty = 2)
  lines(x = year.raw, y =ret.df$upper,  col = cols[3], lwd = 2, lty = 2)
}




#### MODEL 1B ####

Xb <- cbind(1, gulls[, 'year'])
```

```r
Bb <- t(as.matrix(combres.1b))
epsb <- 0

ret.dfb <- sampler.1e.2(X = Xb, B=Bb, eps = 0)


#### MODEL 1C ####
Sc <- nrow(combres.1c)


Xc  <- cbind(1, gulls[, 'year'])
Bc  <- t(as.matrix(combres.1c[, c('b0', 'b1')]))
eps <- rbind(rnorm(Sc, 0, sd = combres.1c[, 'sigma.epsilon']), NULL)

ret.dfc <- sampler.1e.2(Xc, Bc, eps)


#### MODEL 1D ####

Sd <- nrow(combres.1d)

Xd   <- as.matrix(cbind(1, gulls[, c('year', 'yellowlegged')]))
Bd   <- t(as.matrix(combres.1d[, c('b0', 'b1', 'b2')]))
epsd <- rbind(rnorm(Sd, 0, sd = combres.1d[, 'sigma.epsilon']), NULL)

ret.dfd <- sampler.1e.2(Xd, Bd, epsd)

#### PLOTS ####

cex <- 2

png('Q1/Q1eFinalPlots.png', width = 1200, height = 500)
par(mfrow = c(1,3),
    mar = c(5, 4,4,2)+ 1)
plotter.1e(ret.df = ret.dfb, main = 'Audouin ~ Year', cex = cex)
title(ylab = 'Audouin Abundance', cex.lab = 1.7)
plotter.1e(ret.df = ret.dfc, main = 'Audouin ~  Year + variance', cex = cex)
plotter.1e(ret.df = ret.dfd, main = 'Audouin ~ Year + Yellowlegged + variance', cex = cex)
par(mfrow=c(1,1))
dev.off()




#### REPORT FIGURES ####

# Capitalise first letter of a string
capitalise <- function(x) {paste(toupper(substr(x,1,1)), substr(x,2,nchar(x)),sep='')}

df1 <- cbind.data.frame(ret.dfb, ret.dfc, ret.dfd)

model.ind <- paste(c(rep('1b', 3), rep('1c', 3), rep('1d', 3)))

```

```r
df1.names <- colnames(df1)
df1.names <- paste(df1.names, model.ind)
colnames(df1) <- capitalise(df1.names)
df1 <- as.data.frame(round(as.matrix(df1), 0))
rownames(df1) <- year.raw

png('Q1/Q1eResults.png', width = 650, height = 600)
grid.table(df1)
dev.off()
```

B.6 1f

```
#### DIC ####

# No need to update, already burned in.

## 16m


start <- Sys.time()
dic.1b <-dic.samples(model = m1.b, n.iter = 20000, type='pD')
dic.1c <- dic.samples(model = m1.c, n.iter = 1000000, type='pD')
dic.1d <- dic.samples(model = m1.d, n.iter = 1400000, type='pD')
end <- Sys.time()
runtime <- end-start
runtime


c1 <- sum(dic.1b$deviance) + sum(dic.1b$penalty)
c2 <- sum(dic.1c$deviance) + sum(dic.1c$penalty)
c3 <- sum(dic.1d$deviance) + sum(dic.1d$penalty)

dic.mat <- cbind(c1, c2, c3)
colnames(dic.mat) <- c('Model 1B', 'Model 1C', 'Model 1D')
rownames(dic.mat) <- 'DIC'
dic.mat <- round(dic.mat, 2)
dic.df <- as.data.frame(dic.mat)

png('Q1/Q1fDIC.png', width = 250, height = 50)
grid.table(dic.df)
dev.off()
```

C Appendix: Question 2

C.1 2a

```
### Funcs from Q1 ###

demean <- function(x) {x - mean(x)}


#### Load ####

require(rjags)
require(runjags)
require(gridExtra)
require(corrplot)
require(dplyr)


cows <- read.csv('Q2/Data/cows.csv')

# -- Correlation between variables of farm environment

### To avoid inflation of correlation, get unique
env.names <- c('temp', 'rain', 'permeab', 'hight', 'slope')
env.df <- cows[, env.names]
env.df.unique <- distinct(env.df)

png('Q2/Q2aEDA.png', width = 930, height = 600)
par(mfrow = c(3, 3))
## Correlation Plot ##
corrplot(corr = cor(env.df.unique), type = 'lower',method = 'number', order = 'hclust')

# -- Farm barplot data

barplot.2('farmID', xlab = 'Farm ID', srt = 90)

# -- Age barplot data

barplot.2('age', xlab = 'Age', srt = 90)

# -- Boxplots
# Remove cowID, farmID, and parasite for boxplot variables
var.names <- colnames(cows)[!(colnames(cows) %in% c('cowID', 'farmID', 'parasite'))]
boxplot.mw(cows, 'parasite', var.names = var.names)
par(mfrow = c(1,1))
dev.off()
```

C.2 2b

```
## ============================================================================##
##################################### 2b: Mod ###################################
## ============================================================================##


#### Extract as vectors ####

para <- cows$parasite

### Fixed ###
# Specific #
age    <- demean(cows$age)

# Environment #
temp   <- demean(cows$temp)
rain   <- demean(cows$rain)
perm   <- demean(cows$permeab)
height <- demean(cows$hight)
slop   <- demean(cows$slope)

# Random
farm <- cows$farmID


#### Data ####
## Dataset ##
n      <- nrow(cows)
J      <- max(farm)
para   <- para
age    <- age
temp   <- temp
rain   <- rain
perm   <- perm
height <- height
slop   <- slop
farm   <- farm

## Prior ##

### Same priors for each beta
beta.mu  <- 0
beta.tau <- 0.01

## Hyperpriors ##
#  Priors on the random parameter alpha
sig.alpha.ub <- 20

## DATA LIST ##

data <- list(n = n, J = J,                                          # Loop idx
             para = para, age = age, temp = temp, rain = rain,      # Covariates
             perm = perm, height = height, slop = slop,
```

```
            farm = farm,
            beta.mu = beta.mu, beta.tau = beta.tau,                    # Priors
            sig.alpha.ub = sig.alpha.ub) # Hyperpriors


#### MODEL ####
modstr.2b <- "model{

  # Likelihood
  for (i in 1:n) {
     para[i] ~ dbern(p[i])
     # alpha is the random farm-specific intercept.
     logit(p[i]) = b0 + alpha[farm[i]] + b1*age[i] + b2*temp[i] + b3*rain[i] +
                   b4*perm[i] + b5*height[i] + b6*slop[i]
  }

  # Priors
  b0 ~ dnorm(beta.mu, beta.tau)
  b1 ~ dnorm(beta.mu, beta.tau)
  b2 ~ dnorm(beta.mu, beta.tau)
  b3 ~ dnorm(beta.mu, beta.tau)
  b4 ~ dnorm(beta.mu, beta.tau)
  b5 ~ dnorm(beta.mu, beta.tau)
  b6 ~ dnorm(beta.mu, beta.tau)

  for (j in 1:J){
    alpha[j] ~ dnorm(0, tau.alpha)
  }

  # Hyperpriors #
  sig.alpha ~ dunif(0, sig.alpha.ub)
  tau.alpha = pow(sig.alpha, -2)


}"

m.2b <- jags.model(textConnection(modstr.2b), data = data, n.chains = 3)


var.names <- c('b0','b1', 'b2', 'b3', 'b4', 'b5', 'b6', 'alpha', 'mu.alpha', 'sig.alpha')



update(m.2b, 50000)


# 12m: 150000 iterations
# 8.7m: 125000?
start_time <- Sys.time()

res.2b <- coda.samples(m.2b, variable.names = var.names, n.iter = 125000, thin = 100)

end_time <- Sys.time()
```

```r
end_time - start_time

## Combine ##
# From runjags
combres.2b <- combine.mcmc(res.2b)



#### CHECK HOW MUCH TO THIN ####
# only run if this is 1

runthincheck <- 0

if (runthincheck == 1){

  thincheck(res.2b, 'b2', 500, 50, dim(res.2b[[1]])[1])
abline(v = c(100, 200, 300, 400, 500), col = c('red', 'blue', 'green', 'purple', 'pink'))

}







#### CONVERGENCE ####
gelman.diag(res.2b) # All 1, upper CI 1.02
effectiveSize(combres.2b) # Minimum of 1800

png('Q2/Q2bTrace.png', width = 1200, height = 800)
par(mfrow = c(5, 6))
traceplot(res.2b)
par(mfrow=c(1,1))
dev.off()



#### RESULTS ####

restab.2b <- results.table(combres.2b)
png('Q2/Q2bResults.png', width = 800, height = 600)
grid.table(restab.2b)
dev.off()


#### Further ####

mean(combres.2b[, 'b1'] > 0)
```

C.3 2c

```
#### TEMP REMOVED MODEL STRING ####

modstr.2c.rtemp <- "model{

  # Likelihood
   for (i in 1:n) {
para[i] ~ dbern(p[i])
# alpha is the random farm-specific intercept.
logit(p[i]) = b0 + alpha[farm[i]] + b1*age[i] +
             #b2*temp[i] +
              b3*rain[i] + b4*perm[i] + b5*height[i] + b6*slop[i]
}

# Priors
b0 ~ dnorm(beta.mu, beta.tau)
b1 ~ dnorm(beta.mu, beta.tau)
#b2 ~ dnorm(beta.mu, beta.tau)
b3 ~ dnorm(beta.mu, beta.tau)
b4 ~ dnorm(beta.mu, beta.tau)
b5 ~ dnorm(beta.mu, beta.tau)
b6 ~ dnorm(beta.mu, beta.tau)

for (j in 1:J){
alpha[j] ~ dnorm(0, tau.alpha)
}

# Hyperpriors #
sig.alpha ~ dunif(0, sig.alpha.ub)
tau.alpha = pow(sig.alpha, -2)

}"

#### TEMP, RAIN REMOVED MODEL STRING ####

modstr.2c.rtemprain <- "model{


  # Likelihood
   for (i in 1:n) {
para[i] ~ dbern(p[i])
# alpha is the random farm-specific intercept.
logit(p[i]) = b0 + alpha[farm[i]] + b1*age[i] +
             #b2*temp[i] + b3*rain[i] +
              b4*perm[i] + b5*height[i] + b6*slop[i]
}

# Priors
b0 ~ dnorm(beta.mu, beta.tau)
b1 ~ dnorm(beta.mu, beta.tau)
#b2 ~ dnorm(beta.mu, beta.tau)
#b3 ~ dnorm(beta.mu, beta.tau)
b4 ~ dnorm(beta.mu, beta.tau)
```

```
b5 ~ dnorm(beta.mu, beta.tau)
b6 ~ dnorm(beta.mu, beta.tau)

for (j in 1:J){
alpha[j] ~ dnorm(0, tau.alpha)
}

# Hyperpriors #
sig.alpha ~ dunif(0, sig.alpha.ub)
tau.alpha = pow(sig.alpha, -2)

}"

#### TEMP, HEIGHT REMOVED MODEL STRING ####

modstr.2c.rtempheight <- "model{


  # Likelihood
   for (i in 1:n) {
para[i] ~ dbern(p[i])
# alpha is the random farm-specific intercept.
logit(p[i]) = b0 + alpha[farm[i]] + b1*age[i] +
             #b2*temp[i] +
              b3*rain[i] + b4*perm[i] +
             #b5*height[i] +
              b6*slop[i]
}

# Priors
b0 ~ dnorm(beta.mu, beta.tau)
b1 ~ dnorm(beta.mu, beta.tau)
#b2 ~ dnorm(beta.mu, beta.tau)
b3 ~ dnorm(beta.mu, beta.tau)
b4 ~ dnorm(beta.mu, beta.tau)
#b5 ~ dnorm(beta.mu, beta.tau)
b6 ~ dnorm(beta.mu, beta.tau)

for (j in 1:J){
  alpha[j] ~ dnorm(0, tau.alpha)
}

# Hyperpriors #
sig.alpha ~ dunif(0, sig.alpha.ub)
tau.alpha = pow(sig.alpha, -2)

}"

#### MODEL INITIALISATION ####

## Note: Should be warnings about unused variables in data

m2c.rtemp       <- jags.model(textConnection(modstr.2c.rtemp),       data = data, n.chains = 2)
```

```
106
107 m2c.rtemprain   <- jags.model(textConnection(modstr.2c.rtemprain),   data = data, n.chains = 2)
108
109 m2c.rtempheight <- jags.model(textConnection(modstr.2c.rtempheight), data = data, n.chains = 2)
110
111 #### BURN IN ####
112 update(m2c.rtemp,        20000)
113 update(m2c.rtemprain,    20000)
114 update(m2c.rtempheight,  20000)
115
116
117 #### DIC SAMPLES ####
118
119 n.iter <- 100000 ## 18 minutes
120
121 start <- Sys.time()
122 dic.full        <- dic.samples(m.2b,            n.iter = n.iter)
123 dic.rtemp       <- dic.samples(m2c.rtemp,       n.iter = n.iter)
124 dic.rtemprain   <- dic.samples(m2c.rtemprain,   n.iter = n.iter)
125 dic.rtempheight <- dic.samples(m2c.rtempheight, n.iter = n.iter)
126 end <- Sys.time()
127 end-start
128
129
130
131 dic.full.val        <- sum(dic.full$deviance)        + sum(dic.full$penalty)
132 dic.rtemp.val       <- sum(dic.rtemp$deviance)       + sum(dic.rtemp$penalty)
133 dic.rtemprain.val   <- sum(dic.rtemprain$deviance)   + sum(dic.rtemprain$penalty)
134 dic.rtempheight.val <- sum(dic.rtempheight$deviance) + sum(dic.rtempheight$penalty)
135
136 ## Results matrix
137 dic.all <- cbind(dic.full.val, dic.rtemp.val, dic.rtemprain.val, dic.rtempheight.val)
138 colnames(dic.all) <- c('Full Model', 'Temp Removed', 'Temp, Rain Removed', 'Temp, Height Removed'
    )
139 rownames(dic.all) <- c('DIC')
140 dic.all <- round(dic.all, 2)
141 dic.all <- as.data.frame(dic.all)
142
143
144
145 png('Q2/Q2cDIC.png', width = 500, height = 50)
146 grid.table(dic.all)
147 dev.off()
148
149 #### PART 2, PARAMETERS OF CHOSEN MODEL ####
150
151 ## Recompile model with 3 chains
152 m2c.rtempheight   <- jags.model(textConnection(modstr.2c.rtempheight), n.chains = 3, data = data)
153
154 ## Burn-in
155 update(m2c.rtempheight, 50000)
156
157 ## Determine variables to keep (remove b2 and b5)
```

```r
var.names <- c('b0', 'b1', 'b2', 'b3', 'b4', 'b5', 'b6', 'alpha', 'sig.alpha')
var.names.rtempheight <- var.names[!(var.names %in% c('b2', 'b5'))]


## Generate samples

start <- Sys.time()
res.2c <- coda.samples(m2c.rtempheight, var.names.rtempheight, n.iter = 150000, thin = 50)
end <- Sys.time()
end - start

combres.2c <- combine.mcmc(res.2c)


#### CONVERGENCE ####
## Traceplots
png('Q2/Q2cTrace.png', width = 1200, height=800)
par(mfrow = c(4, 6))
traceplot(res.2c)
par(mfrow=c(1,1))
dev.off()


effectiveSize(res.2c)
gelman.diag(res.2c)
gelman.plot(res.2c)

#### RESULTS ####

restab.2c <- results.table(combres.2c)
png('Q2/Q2cResults.png', width = 800, height = 600)
grid.table(restab.2c)
dev.off()

#### RESULTS EVALUATION ####
b6.2c <- combres.2c[, 'b6']
mean(b6.2c < 0.01 & b6.2c > -0.01)

#### RESULTS COMPARISON ####
restab.2b <- results.table(combres.2b)
restab.2b <- as.data.frame(restab.2b)
restab.2c <- as.data.frame(restab.2c)

### JOIN ###
joined <- merge(restab.2c, restab.2b, by = 0)

### Meandif

meandif <- cbind.data.frame(joined$Mean.x, joined$Mean.y)
meandif
View(meandif)
```

C.4 2d

```
### Get common function inputs for both farms

# Data matrix and coefficients
# 1 (intercept), age, rain, permeability, slope
# age = 0 since assuming mean age (mean(age) - mean(age)) since demeaned variable
X <- cbind(1, age=0, rain, perm, slop)
B <- combres.2c[, c('b0', 'b1', 'b3', 'b4', 'b6')]
Bt <- t(as.matrix(B))

#### Farm 1 ####
ID <- 1

# Get farm 1 matrix
X.2d.1 <- X[cows$farmID == ID, ]
# Extract 1st row, since all values same for a given farm
xt.2d.1 <- t(as.matrix(X.2d.1[1, ]))

# Extract farm-specific effect.
a.2d.1 <- combres.2c[, 'alpha[1]']

probs.2d.1 <- farmprobs(xt = xt.2d.1, Bt = Bt, alpha = a.2d.1)
post.2d.1  <- results.post(1, probs.2d.1, 3)

#### Farm 6 ####

ID <- 6

# Get farm 1 matrix
X.2d.6 <- X[cows$farmID == ID, ]
# Extract 1st row, since all values same for a given farm
xt.2d.6 <- t(as.matrix(X.2d.6[1, ]))

# Extract farm-specific effect.
a.2d.6 <- combres.2c[, 'alpha[6]']

probs.2d.6 <- farmprobs(xt = xt.2d.6, Bt = Bt, alpha = a.2d.6)
post.2d.6  <- results.post(6, probs.2d.6, 3)

#### SUMMARISE RESULTS ####

## Combined posterior summary ##

Q2d.results <- rbind(post.2d.1, post.2d.6)
rownames(Q2d.results) <- paste(rownames(Q2d.results), '(Rand)')
Q2d.results <- Q2d.results[, c('Mean', '95% CI', 'P(Epidemic)')]

##### REPORT RESULTS #####
png('Q2/Q2dResultsFARM.png', height = 80, width = 300)
grid.table(Q2d.results)
dev.off()

## Histograms ##
```

```
53
54
55 png('Q2/Q2dHists.png', width = 800, height = 300)
56 par(mfrow = c(1, 2))
57 hist(probs.2d.1, main = 'Histogram of Farm 1 Proportions', xlab = 'Farm 1 Proportion')
58 abline(v = c(post.2d.1[, '2.5%'], post.2d.1[, '97.5%'], post.2d.1[, 'Mean']),
59        col = c('Red', 'Red', 'Blue'), lwd = 2)
60
61 hist(probs.2d.6, main = 'Histogram of Farm 6 Proportions', xlab = 'Farm 6 Proportion')
62 abline(v = c(post.2d.6[, '2.5%'], post.2d.6[, '97.5%'], post.2d.6[, 'Mean']),
63        col = c('Red', 'Red', 'Blue'), lwd = 2)
64 par(mfrow = c(1,1))
65 dev.off()
```

C.5 2e

```
modstr.2e <- "model{


# Likelihood
  for (i in 1:n) {
      para[i] ~ dbern(p[i])
      # alpha is the FIXED farm-specific intercept.
      logit(p[i]) = alpha[farm[i]] + b1*age[i] +
      #b2*temp[i] +
      b3*rain[i] + b4*perm[i] +
      #b5*height[i] +
      b6*slop[i]
  }

  # Priors
  b1 ~ dnorm(beta.mu, beta.tau)
  #b2 ~ dnorm(beta.mu, beta.tau)
  b3 ~ dnorm(beta.mu, beta.tau)
  b4 ~ dnorm(beta.mu, beta.tau)
  #b5 ~ dnorm(beta.mu, beta.tau)
  b6 ~ dnorm(beta.mu, beta.tau)

  for (j in 1:J){
    alpha[j] ~ dnorm(beta.mu, beta.tau)
  }

}"



mod.2e <- jags.model(textConnection(modstr.2e), data = data, n.chains = 3)

update(mod.2e, 100000)

var.names.2e <- c('b0', 'b1', 'b3', 'b4', 'b6', 'alpha')


start <- Sys.time()
res.2e <- coda.samples(mod.2e, variable.names = var.names.2e, n.iter = 400000,
                       thin = 200)
Sys.time() - start

combres.2e <- combine.mcmc(res.2e)

#### CONVERGENCE ####
effectiveSize(res.2e)

png('Q2/Q2eTrace.png', width = 1200, height=800)
par(mfrow = c(4, 6))
traceplot(res.2e)
par(mfrow=c(1,1))
dev.off()
```

```
53
54 gelman.diag(res.2e)
55
56
57 #### RESULTS ####
58 restab.2e <- results.table(combres = combres.2e)
59
60
61 ##### REPORT RESULTS OF PARAMETERS #####
62 png('Q2/Q2eResults.png', width = 800, height = 600)
63 grid.table(restab.2e)
64 dev.off()
65
66
67
68 #### FARM SPECIFIC ####
69
70 ## Recall objects of interest:
71 # X: Same rows are used, so just copy
72 # No b0, so remove the first element with '-1' as column index
73 xt.2e.1 <- t(as.matrix(xt.2d.1[-1]))
74 xt.2e.6 <- t(as.matrix(xt.2d.6[-1]))
75
76 # B, alpha: Need to extract new parameters
77 B  <- combres.2e[, c('b1', 'b3', 'b4', 'b6')]
78 Bt <- t(as.matrix(B))
79
80 #### FARM 1
81 ID <- 1
82
83 # Extract farm-specific effect.
84 a.2e.1 <- combres.2e[, 'alpha[1]']
85
86 probs.2e.1 <- farmprobs(xt = xt.2e.1, Bt = Bt, alpha = a.2e.1)
87 post.2e.1  <- results.post(1, probs.2e.1, 5)
88
89
90 #### FARM 6
91
92 ID <- 6
93
94 # Extract farm-specific effect.
95 a.2e.6 <- combres.2e[, 'alpha[6]']
96
97 probs.2e.6 <- farmprobs(xt = xt.2e.6, Bt = Bt, alpha = a.2e.6)
98 post.2e.6  <- results.post(6, probs.2e.6, 5)
99
100 #### SUMMARISE RESULTS ####
101
102 ## Combined posterior summary ##
103
104 Q2e.results <- rbind(post.2e.1, post.2e.6)
105 rownames(Q2e.results) <- paste(rownames(Q2e.results), '(Fixed)')
```

```
Q2e.results <- Q2e.results[, c('Mean', '95% CI', 'P(Epidemic)')]

##### REPORT RESULTS #####
png('Q2/Q2eResultsFARM.png', height = 80, width = 340)
grid.table(Q2e.results)
dev.off()

## Histograms ##


png('Q2/Q2eHists.png', width = 800, height = 300)
par(mfrow = c(1, 2))
hist(probs.2e.1, main = 'Histogram of Farm 1 Proportions', xlab = 'Farm 1 Proportion')
abline(v = c(post.2e.1[, '2.5%'], post.2e.1[, '97.5%'], post.2e.1[, 'Mean']),
       col = c('Red', 'Red', 'Blue'), lwd = 2)

hist(probs.2e.6, main = 'Histogram of Farm 6 Proportions', xlab = 'Farm 6 Proportion')
abline(v = c(post.2e.6[, '2.5%'], post.2e.6[, '97.5%'], post.2e.6[, 'Mean']),
       col = c('Red', 'Red', 'Blue'), lwd = 2)
par(mfrow = c(1,1))
dev.off()
```

D Appendix: Question 3

```
#### MODEL STRING ####

modstr.3 <- "model{


# Likelihood
for (i in 1:n) {
para[i] ~ dnorm(p[i], para.tau)

  p[i] = b0 + b1*age[i] +
                b3*rain[i] +
                b4*perm[i] +
                b6*slop[i]

}

# Priors
b0 ~ dnorm(beta.mu, beta.tau)
b1 ~ dnorm(beta.mu, beta.tau)
b3 ~ dnorm(beta.mu, beta.tau)
b4 ~ dnorm(beta.mu, beta.tau)
b6 ~ dnorm(beta.mu, beta.tau)
para.tau ~ dgamma(0.01, 0.01)

}"

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

### DIC ###
update(m.3, 50000)
dic.3 <- dic.samples(m.3, 150000)

dic.3


#### MODEL ####
res.3 <- coda.samples(m.3, c('b0', 'b1',  'b3', 'b4', 'b6'), n.iter = 50000, thin = 10)

combres.3 <- combine.mcmc(res.3)


#### CONVERGENCE ####

png('Q2/Q3Trace.png', width = 700, height = 400)
par(mfrow = c(2, 3))
traceplot(res.3)
par(mfrow = c(1,1))
dev.off()

#### RESULTS ####
restab.3 <- results.table(combres.3, dig = 4)
```

```
52
53 png('Q2/Q3Results.png', width = 750, height = 150)
54 grid.table(restab.3)
55 dev.off()
56
57 ### COMP ###
58
59 mean(combres.3[, 'b0'] > 0)
60 mean(combres.3[, 'b1'] > 0)
61 mean(combres.3[, 'b3'] > 0)
62 mean(combres.3[, 'b4'] < 0)
63 mean(combres.3[, 'b6'] < 0)
64
65 mean(combres.2c[, 'b0'] > 0)
66 mean(combres.2c[, 'b1'] > 0)
67 mean(combres.2c[, 'b3'] > 0)
68 mean(combres.2c[, 'b4'] < 0)
69 mean(combres.2c[, 'b6'] < 0)
70
71
72 ### PREDICTIONS ###
73 X <- cbind(1, age, rain, perm, slop)
74 Bt <- t(as.matrix(combres.3))
75
76 lc <- X%*%Bt
77 pr <- apply(lc, 1, mean)
78 table(pr>0.5, para)
```