

PHP2530 Problem Set 3

Alyson Singleton

3/24/2020

Chapter 5 Problem 3

5.3 Part A

Table 1: Educational Testing Example

	Pr(best)	A	B	C	D	E	F	G	H
A	0.265	0.000	0.675	0.700	0.630	0.720	0.715	0.515	0.635
B	0.095	0.325	0.000	0.530	0.510	0.615	0.580	0.345	0.465
C	0.095	0.300	0.470	0.000	0.425	0.555	0.520	0.325	0.435
D	0.085	0.370	0.490	0.575	0.000	0.635	0.595	0.375	0.495
E	0.050	0.280	0.385	0.445	0.365	0.000	0.475	0.245	0.380
F	0.050	0.285	0.420	0.480	0.405	0.525	0.000	0.255	0.415
G	0.220	0.485	0.655	0.675	0.625	0.755	0.745	0.000	0.610
H	0.140	0.365	0.535	0.565	0.505	0.620	0.585	0.390	0.000

5.3 Part B

Table 2: Educational Testing Example w Tau \rightarrow Inf

	Pr(best)	A	B	C	D	E	F	G	H
A	0.560	0.000	0.860	0.900	0.875	0.950	0.935	0.720	0.820
B	0.045	0.140	0.000	0.695	0.530	0.750	0.680	0.265	0.500
C	0.055	0.100	0.305	0.000	0.390	0.490	0.465	0.185	0.335
D	0.035	0.125	0.470	0.610	0.000	0.685	0.645	0.240	0.445
E	0.005	0.050	0.250	0.510	0.315	0.000	0.420	0.085	0.315
F	0.015	0.065	0.320	0.535	0.355	0.580	0.000	0.145	0.355
G	0.200	0.280	0.735	0.815	0.760	0.915	0.855	0.000	0.635
H	0.085	0.180	0.500	0.665	0.555	0.685	0.645	0.365	0.000

5.3 Part C

When we set $\tau = \infty$, the model is less conservative. When τ is given a finite value (i.e. we create a the full hierarchical model), said model is able to recognize and implement the fact that there is evidence in the data that the schools are fairly similar in their effectiveness. For example, the probability that School A is the best increases from 0.26 to 0.56—it predicts that it is much more likely. This also holds for part (ii). For example, the probability that School A's program is better than School H under the full hierarchical model is 0.635, whereas it increases to 0.82 under the $\tau = \infty$ model.

5.3 Part D

If $\tau = 0$ then the probabilities are all “degenerate” in that they are all 0. This is because by setting τ equal to zero, we are effectively telling the model that all of the school effects are the same and no school is better or worse than the other.

Chapter 5 Problem 4

5.4 Part A

$\theta_1, \dots, \theta_{2J}$ are indeed exchangeable. To demonstrate this, we can consider all of the permutations, p , of the indices, $1, \dots, 2J$, write their joint prior distribution and investigate.

$$p(\theta_1, \dots, \theta_{2J}) = \binom{2J}{J}^{-1} \sum_p \left(\prod_{j=1}^J N(\theta_{p(j)} | 1, 1) \prod_{j=J+1}^{2J} N(\theta_{p(j)} | -1, 1) \right)$$

When we sum over all of the possible permutations (i.e. all possible p) and normalize by dividing by the number of them, the density ($p(\theta_1, \dots, \theta_{2J})$) is unaffected by the permutations of the indices. We can write it this way because the first set (those from the $N(1,1)$ distribution) is iid and the second (those from the $N(-1,1)$ distribution) is as well. This renders the parameters exchangeable as the prior is invariant to the order of the indices.

5.4 Part B

Pick any i, j from $\{1, \dots, 2J\}$. Consider the relationship of the corresponding θ_i and θ_j . If θ_i is large (therefore positive) then it likely came from the $N(1,1)$ distribution. Because we know that exactly half of the $2J$ parameters came from each of the two distributions, we can recognize, with the knowledge that θ_i is from the $N(1,1)$ distribution, there is a slightly higher probability that θ_j came from the $N(-1,1)$ distribution rather than from the $N(1,1)$ distribution. Therefore, θ_j will likely be negative. I can make a parallel argument for when θ_i is negative, that θ_j will have a slightly higher likelihood probability of being positive. In either of these two scenarios, the covariance of θ_i and θ_j is negative. If the covariance of the two parameters is negative, they cannot be written as a mixture of independent and identically distributed components (a quality referenced in problem 5.5 but that I had trouble writing out myself).

5.4 Part C

We start by recalling de Finetti’s Theorem. It states that in the limit as $J \rightarrow \infty$, any “suitably well-behaved” exchangeable distribution on $(\theta_1, \dots, \theta_J)$ can be expressed as a mixture of independent and identical distributions. This example does not hold water as a counterexample to de Finetti’s Theorem because as $J \rightarrow \infty$, the negative correlation between θ_i and θ_j will approach zero. With an infinite amount of parameters, the need for exactly half of the θ_i ’s to be in each group will no longer impact the selection of the distribution from which each θ_i is drawn. This is another way of saying that they will not only be independent within each of the groups, but also between the two groups.

Chapter 5 Problem 11

5.11 Part A

We wish to find the joint posterior density, $p(\theta, \mu, \tau | y)$. We can write the following:

$$p(\theta, \mu, \tau | y) \propto p(y | \theta, \mu, \tau) p(\theta | \mu, \tau) p(\mu, \tau)$$

We note that y is independent of μ, τ and these two parameters are related to y only through their relationship to θ . This conclusion can be written as: $p(y|\theta, \mu, \tau) = p(y|\theta)$. Recall from 5.3 that $p(y|\theta) \sim \text{Bin}(n_j, \theta_j)$ and that given that $p(\mu, \tau) \propto 1$. Indeed, the problem statement also gives that $\text{logit}(\theta_j) \sim N(\mu, \tau^2)$. Therefore we can rewrite our original statement as:

$$p(\theta, \mu, \tau|y) \propto p(y|\theta)p(\theta|\mu, \tau) * 1$$

$$p(\theta, \mu, \tau|y) \propto p(\theta|\mu, \tau) \prod_{j=1}^J \theta_j^{y_j} (1 - \theta_j)^{n_j - y_j}$$

All that is left for us to find is the new transformation of $p(\theta|\mu, \tau)$ given $\text{logit}(\theta_j) \sim N(\mu, \tau^2)$. We can transform this using the variable transformation technique we learned in class:

$$p_v(v) = |J| p_u(f^{-1}(v))$$

Let's let $v = (\theta_j)$ and $u = \text{logit}(\theta_j)$. Therefore,

$$p_v(v) = \left| \frac{d}{d\theta_j} \text{logit}(\theta_j) \right| p_u(\theta_j)$$

$$p_v(v) = \left| \frac{1}{\theta_j(1 - \theta_j)} \right| \exp\left(-\frac{1}{2} \frac{(\text{logit}(\theta_j) - \mu)^2}{\tau^2}\right)$$

Therefore, our final form of the posterior distribution is:

$$p(\theta, \mu, \tau|y) \propto \prod_{j=1}^J \frac{1}{\theta_j(1 - \theta_j)} \exp\left(-\frac{1}{2} \frac{(\text{logit}(\theta_j) - \mu)^2}{\tau^2}\right) \prod_{j=1}^J \theta_j^{y_j} (1 - \theta_j)^{n_j - y_j}$$

5.11 Part B

The general form integral from 5.4 is as follows:

$$p(\phi|y) = \int p(\theta, \phi|y) d\theta$$

For this question we can rewrite it as:

$$p(\mu, \tau|y) = \int p(\theta, \mu, \tau|y) d\theta$$

This is essentially us attempting to integrate the density function we determined in Part A. We would be able to use the independence of the J samples to separate the inside of the integral into independent factors and look at each of the J integrals individually. However, I see no obvious method to combine any of the sets of terms in the expression through transformations or substitutions.

5.11 Part C

The referenced expression 5.5 is as follows:

$$p(\phi|y) = \frac{p(\theta, \phi|y)}{p(\theta|\phi, y)}$$

We may similarly rewrite it as:

$$p(\mu, \tau|y) = \frac{p(\theta, \mu, \tau|y)}{p(\theta|\mu, \tau, y)}$$

The issue here is our lack of knowledge about $p(\theta|\mu, \tau, y)$. Although we have information about this conditional posterior density, we only have up to proportionality in θ . This knowledge is insufficient to evaluate the above expression because our goal would be to use it to find a density that depends explicitly on μ and τ .

Chapter 5 Problem 13

5.13 Part A

We wish to find the joint posterior distribution: $p(\theta, \alpha, \beta|y)$, we can write this out as as proportional to the product of the likelihood and the joint prior distribution as such:

$$p(\theta, \alpha, \beta|y) \propto (y|\theta, \alpha, \beta)p(\theta|\alpha, \beta)p(\alpha, \beta)$$

Note that I am using y to represent the number of bicycles, y_1, \dots, y_{10} are the number at each of the ten locations, and we let n_1, \dots, n_{10} be the number of total number of vehicles (bicycles and cars) at each of the ten locations. Therefore, from the direction given by the problem statement (where $j = 1, \dots, 10$):

$$p(y_j|\theta_j) \sim \text{Bin}(n_j, \theta_j)$$

$$p(y|\theta, \alpha, \beta) = \prod_{j=1}^{10} \binom{n_j}{y_j} \theta_j^{y_j} (1 - \theta_j)^{n_j - y_j}$$

$$p(\theta_j) \sim \text{Beta}(\alpha, \beta)$$

$$p(\theta|\alpha, \beta) = \prod_{j=1}^{10} \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta_j^{\alpha-1} (1 - \theta_j)^{\beta-1}$$

And, lastly, we need are directed to choose a non-informative prior distribution for the hyperparameters, i.e. $p(\alpha, \beta)$. We will use the distribution derived in the book (5.9, p.110):

$$p(\alpha, \beta) \propto (\alpha + \beta)^{-\frac{5}{2}}$$

Therefore, our final joint posterior distribution follows:

$$p(\theta, \alpha, \beta|y) \propto \prod_{j=1}^{10} \binom{n_j}{y_j} \theta_j^{y_j} (1 - \theta_j)^{n_j - y_j} \prod_{j=1}^{10} \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta_j^{\alpha-1} (1 - \theta_j)^{\beta-1} (\alpha + \beta)^{-\frac{5}{2}}$$

$$p(\theta, \alpha, \beta|y) \propto (\alpha + \beta)^{-\frac{5}{2}} \left(\frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \right)^{10} \prod_{j=1}^{10} \theta_j^{\alpha+y_j-1} (1 - \theta_j)^{\beta+n_j-y_j-1}$$

5.13 Part B

Next we must find the form of the marginal posterior density of the hyperparameters. Recall that:

$$p(\alpha, \beta) = \frac{p(\theta, \alpha, \beta|y)}{p(\theta|\alpha, \beta, y)}$$

Therefore:

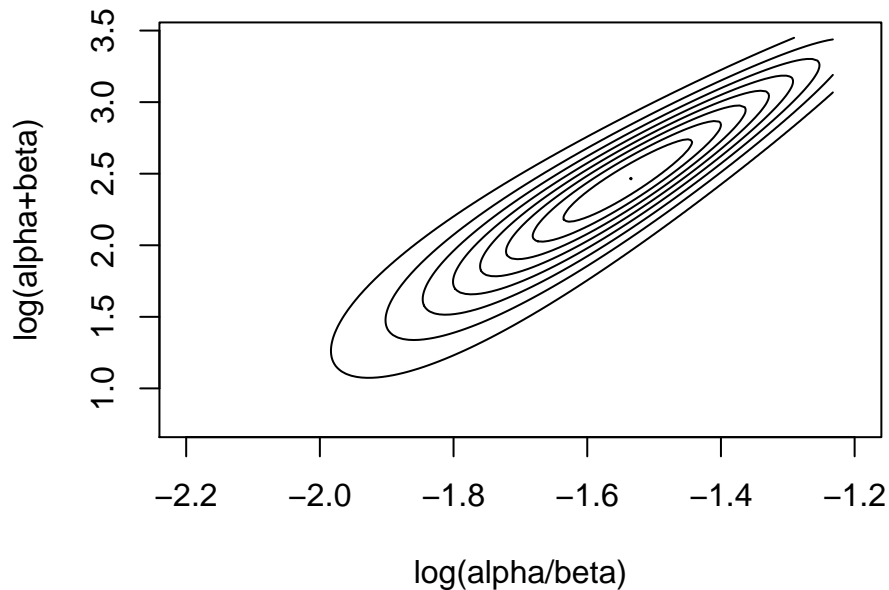
$$p(\alpha, \beta) = \frac{(\alpha + \beta)^{-\frac{5}{2}} \left(\frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \right)^{10} \prod_{j=1}^{10} \theta_j^{\alpha+y_j-1} (1 - \theta_j)^{\beta+n_j-y_j-1}}{\prod_{j=1}^{10} \frac{\Gamma(\alpha + \beta + n_j)}{\Gamma(\alpha + y_j)\Gamma(\beta + n_j - y_j)} \theta_j^{\alpha+y_j-1} (1 - \theta_j)^{\beta+n_j-y_j-1}}$$

Therefore:

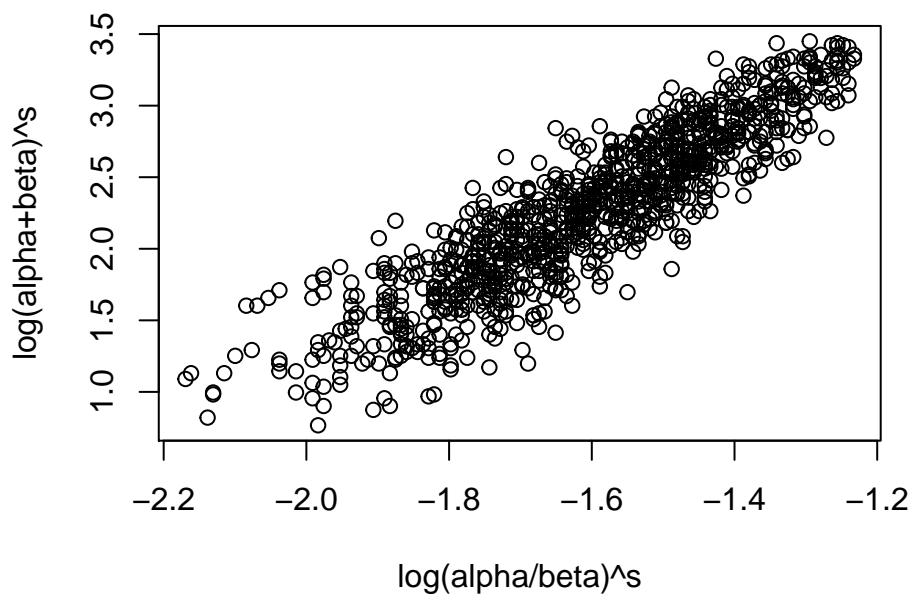
$$p(\alpha, \beta) = (\alpha + \beta)^{-\frac{5}{2}} \left(\frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \right)^{10} \prod_{j=1}^{10} \frac{\Gamma(\alpha + \beta + n_j)}{\Gamma(\alpha + y_j)\Gamma(\beta + n_j - y_j)}$$

Although we cannot simplify this farther, we are able to compute it. See below for a contour plot of joint posterior along with a plot of the simulation draws from the joint posterior distribution of the parameters.

Contour plot of joint posterior



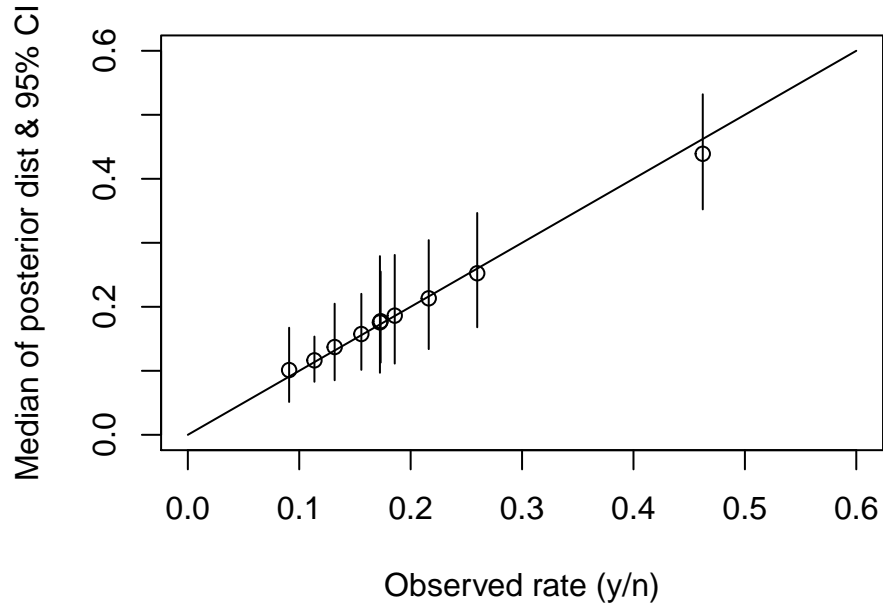
Sample Draws



5.13 Part C

To compare the posterior distributions of the parameters θ_j to the raw proportion in location j plotted them against one another. The inferences from the posterior distribution appear to be quite similar to the raw proportions (you can see that the points fall roughly along the diagonal, as we would hope).

Posterior distribution of biker proportion



5.13 Part D

Our output 95% posterior interval for the average underlying proportion of traffic that is bicycles follows:

Table 3: 5.13 Part D

2.5%	97.5%
0.1526188	0.2442689

5.13 Part E

We look to find a posterior predictive distribution when $n=100$. We generate new values of θ and then use that to generate new estimates of y where $n=100$. Our output 95% posterior interval for the for the number of the 100 vehicles that are bicycles follows:

Table 4: 5.13 Part E

2.5%	97.5%
1	47

Our output 95% CI of [1,47] means we will see 1 to 47 bicycles on a residential street that has bike routes (newly sampled) with the probability of 95%. Notably, it is not very informative as it covers almost all observed values.

5.13 Part F

The plot from part C seems to show that we are fitting the observed data well, leading me to believe that the the beta distribution for the θ_j are reasonable.

Chapter 6 Problem 2

6.2 Part A

I created two test statistics—one for each of the assumptions we would like to investigate. For the first assumption, that we were employing independent Poisson distributions, I used the chi-squared statistic as my test quantity, i.e.:

$$\chi^2 = \frac{(Observed - Expected)^2}{Expected}$$

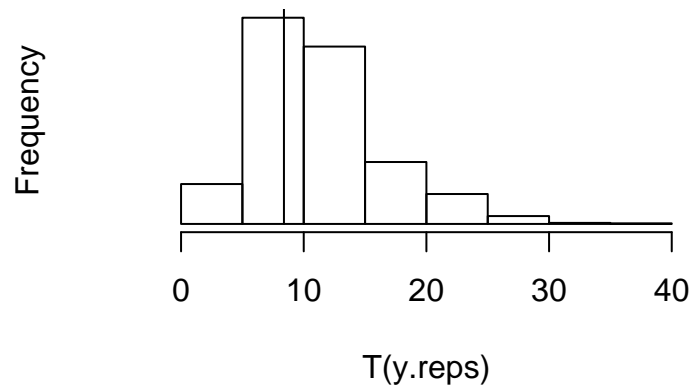
As my data and set up assumes independence, I would expect the observed chi-squared statistic to be similar to the distribution produced by my model.

For the second assumption, that there was no trend over time, I simply calculated the correlation that the values had with time (years). If there was no trend over time we would expect the correlation to center at 0.

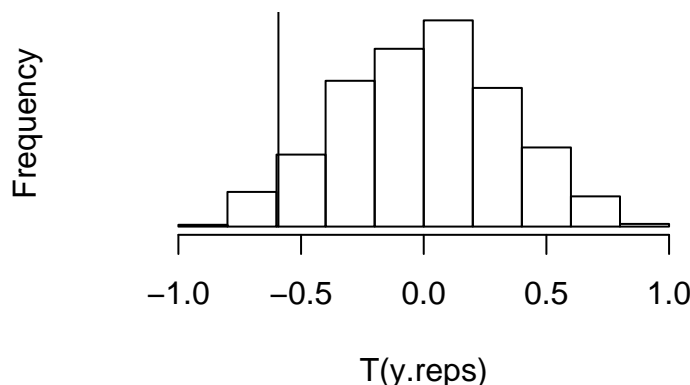
6.2 Part B

For the first model, in which we are modelling the number of fatal accidents, we would expect the assumption of independence to hold and the assumption of no trend to fail. We expect that there to be a downwards trend in the number of accidents—plane travel becoming safer over time. You can see the discrepancies displayed graphically below. The histograms represent the distribution of y^{rep} values and the vertical lines represent the observed values. Both are as expected (independence hold, no trend does not). P-values and discussion surrounding them can be found in a summary table in Part C.

Model 1: Assumption 1

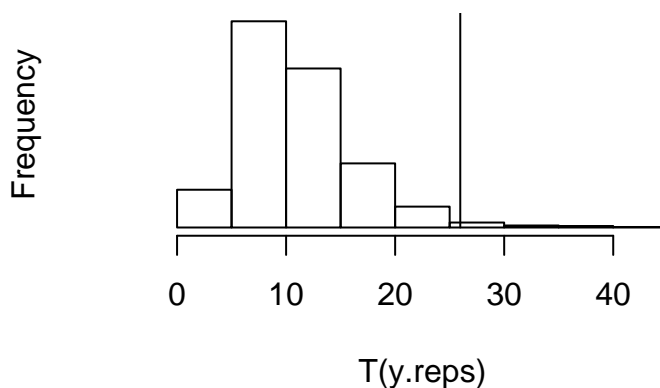


Model 1: Assumption 2

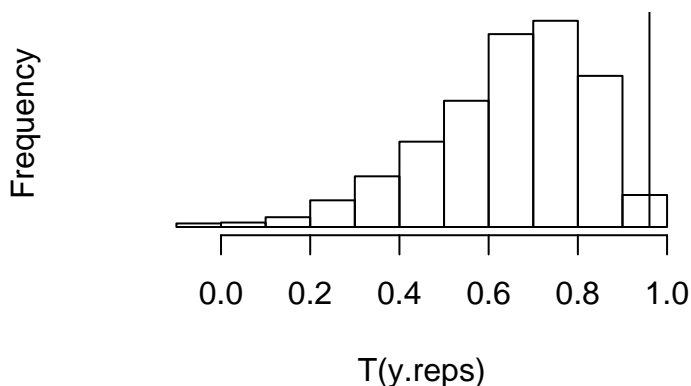


For the second model, in which we are modelling the number of fatal accidents with a rate proportional to passenger miles, we would expect the assumption of independence to hold. Here, the assumption of no trend is hard to predict. It is possible that it will hold given that the increase in safety (i.e. a decrease in accidents) might be balanced out by trend of an increase in overall passenger miles, which give opportunity for more accidents to occur. I am a bit confused here if I should be comparing my test quantities to measures also proportional to passenger miles. I believe that I should be, but that will likely disrupt the baseline assumption of “no trend,” correct? Please note that I went ahead and adjusted everything to be proportional to the passenger miles flown that year (I did this for model 4 as well). With this in mind, the assumption of independence under this model, unexpectedly, fails. Additionally, the no trend assumption also appears inappropriate given the observed correlation value (which is quite positive, indicating that the increase in amount of miles flown likely overtook the potential increases in safety). Again, the observed P-values and discussion surrounding them can again be found in a summary table in Part C.

Model 2: Assumption 1

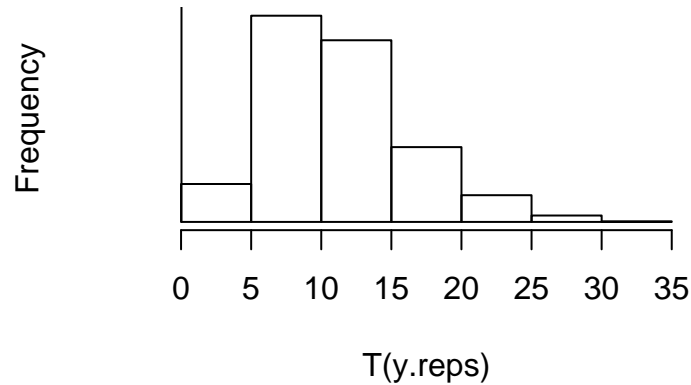


Model 2: Assumption 2

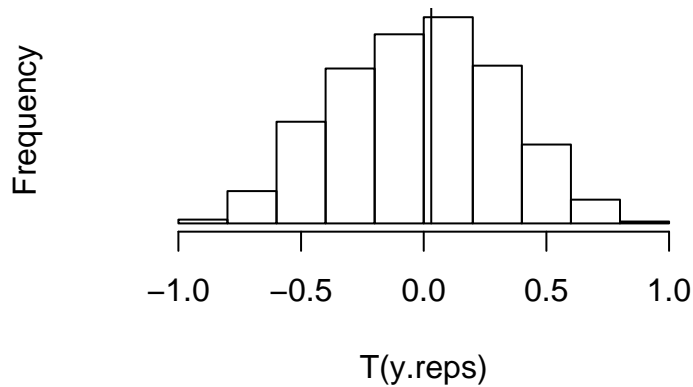


For the third model, in which we are modelling the number of passenger deaths, we would not expect the assumption of independence to hold (passenger deaths happen in clusters). Here, the assumption of no trend is hard to predict. It could potentially decrease (again due to safety), but the clustering adds an interesting component that I don’t have great intuition about. Independence does not hold, as we expected. Surprisingly, the assumption of no trend also appears to hold! P-values and discussion surrounding them can again be found in a summary table in Part C.

Model 3: Assumption 1

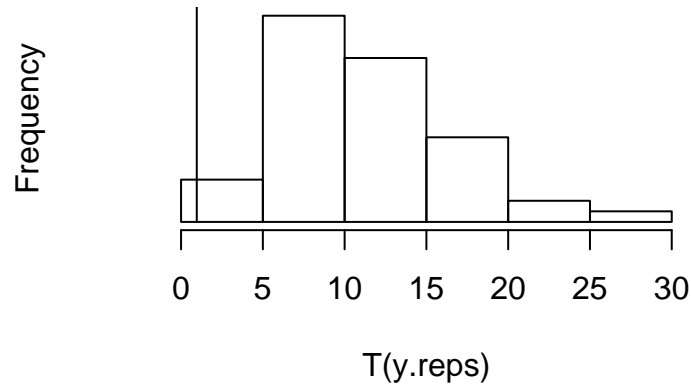


Model 3: Assumption 2

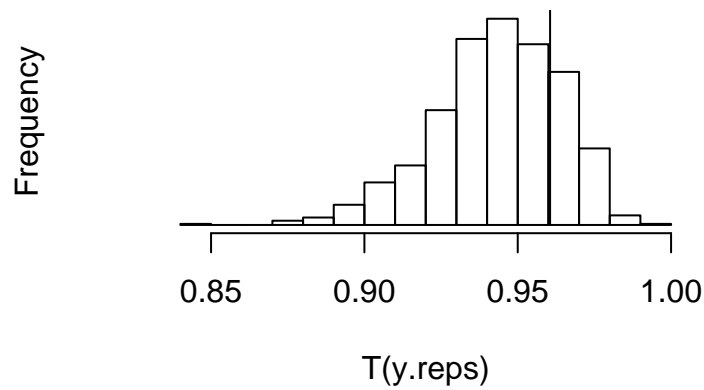


For the fourth model, in which we are modelling the number of passenger deaths with a rate proportional to passenger miles, we would not expect the assumption of independence to hold (passenger deaths happen in clusters). Here, the assumption of no trend is hard to predict. Similar to Model 2, it is possible that it will hold given that the increase in safety (i.e. a decrease in accidents) might be balanced out by trend of an increase in overall passenger miles, which give opportunity for more accidents to occur. Independence does not hold, as we expected. The assumption of no trend does not appear to hold but the model appears to reflect this (this result goes along with my confusion discussed with model 2). P-values and discussion surrounding them can again be found in a summary table in Part C.

Model 4: Assumption 1



Model 4: Assumption 2



6.2 Part C

In summary, as expected, models 1 and 2 adhere to the assumption of independence much more than models 3 and 4. The investigation surrounding the assumption of no trend had some unexpected / interesting results that I would like to explore further.

Table 5: P-Value Assumptions Summary

	Assumption 1	Assumption 2
Model 1	0.675	0.958
Model 2	0.016	0.000
Model 3	0.000	0.453
Model 4	0.000	0.218

Chapter 6 Problem 6

6.6 Part A

The posterior distribution of the parameter θ under the assumed model does not change because the likelihood does not change under the new protocol given this set of observed data. I am assuming that there is no

reason to believe that the prior would have changed. If neither the prior nor the likelihood changes, neither will the posterior. The reason that the likelihood does not change is because the data: $\{1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0\}$ satisfies both rules. There are 20 outcomes, and the 13th zero to appear is at the 20th trial. Formally, the new likelihood can be written out with indicator functions as follows:

$$p(y, n|\theta) = \prod_{i=1}^n \theta^{y_i} (1 - \theta)^{1-y_i} \mathbf{1}_{(y_n=0)} \mathbf{1}_{(\sum_{i=1}^{n-1} (1-y_i)=12)}$$

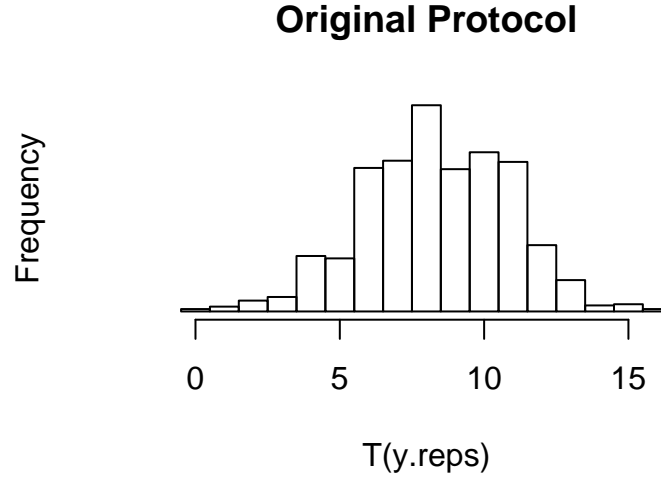
Therefore, under the new protocol and with the given data set, the likelihood would be $\theta^7(1 - \theta)^{13}$, the same as what it would have been in the original set up. For reference, the likelihood in the original set up was as follows:

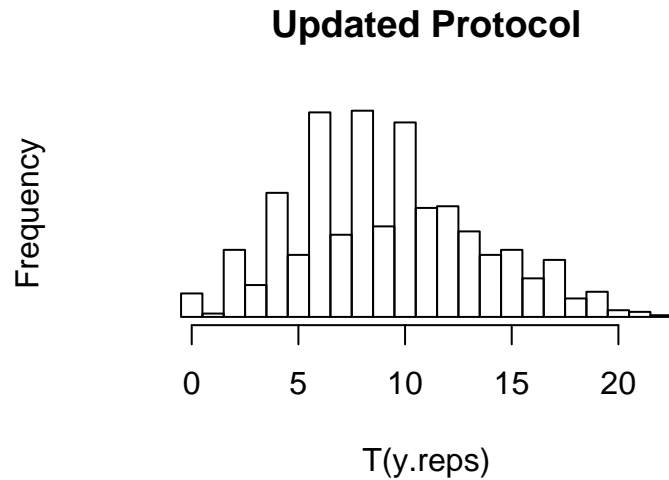
$$p(y|\theta) = \prod_{i=1}^n \theta^{\sum_{i=1}^n y_i} (1 - \theta)^{(n - \sum_{i=1}^n y_i)}$$

And would produce the same expression: $\theta^7(1 - \theta)^{13}$, given the data.

6.6 Part B

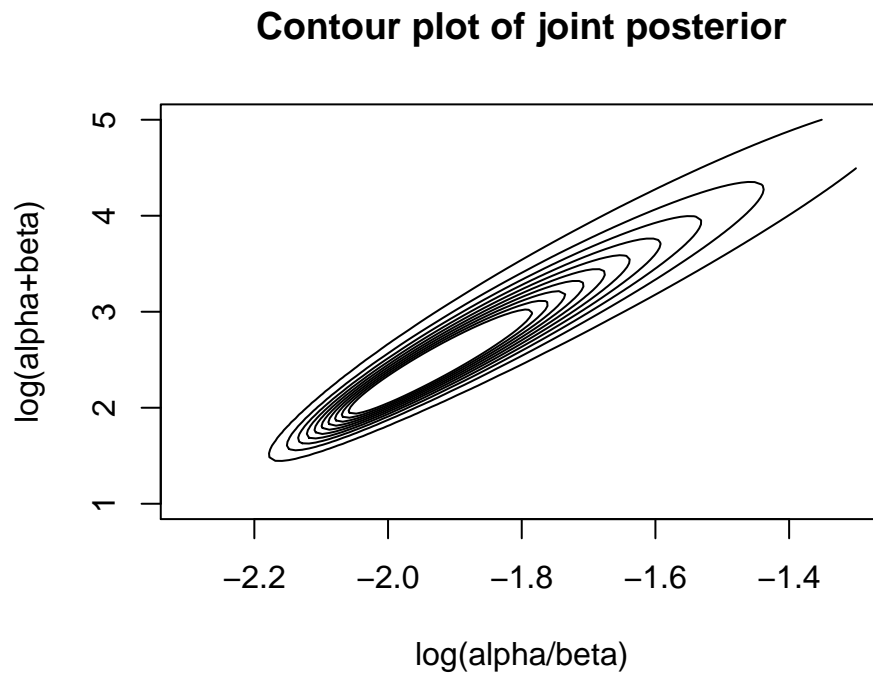
See below for the predictive simulations under both the original protocol and the updated. I notice two main differences between the two. First, the updated protocol produces a much larger spread. This makes sense as the introduction of randomness into the n value itself would produce a larger variance in the number of switches. Secondly, the test quantity appears to be much more likely to be even under the new protocol, especially at lower values. I am interested in why this might be, potentially dependent on the prior distribution of θ ?



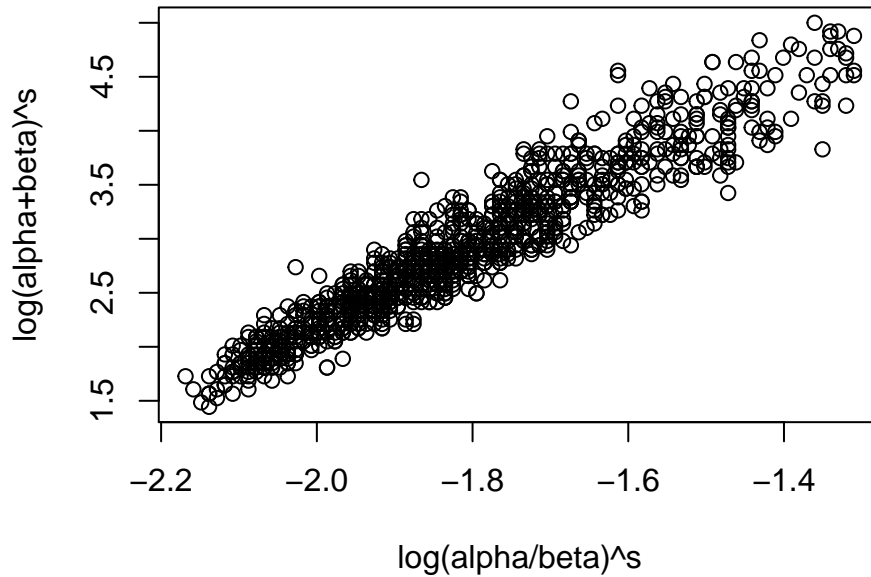


Chapter 6 Problem 9

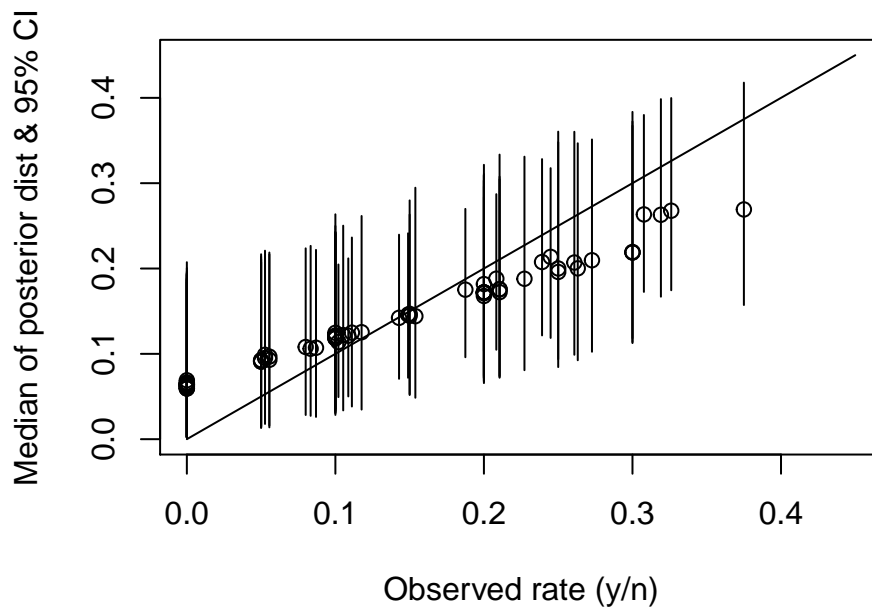
See below for a contour plot of joint posterior along with a plot of the simulation draws from the joint posterior distribution of the parameters. I see that it looks slightly different that the plots in the book. I have parsed through my code multiple times and can't seem to find where I'm going wrong. Luckily the predictions do seem to be consistent with those in the book (the rates are similarly shrunk from their sample point estimates). These are also displayed below for reference.



Sample Draws



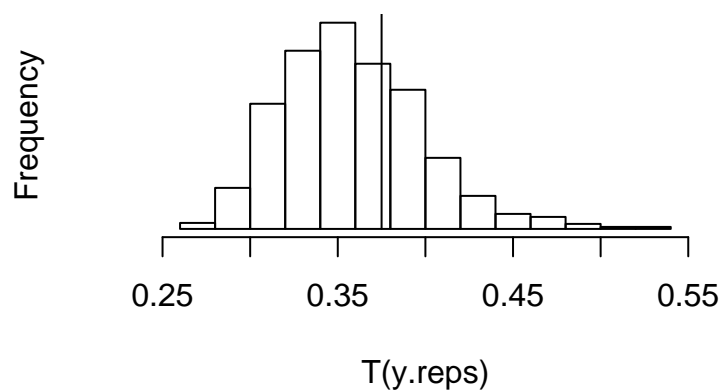
Posterior distribution of rat tumor proportion



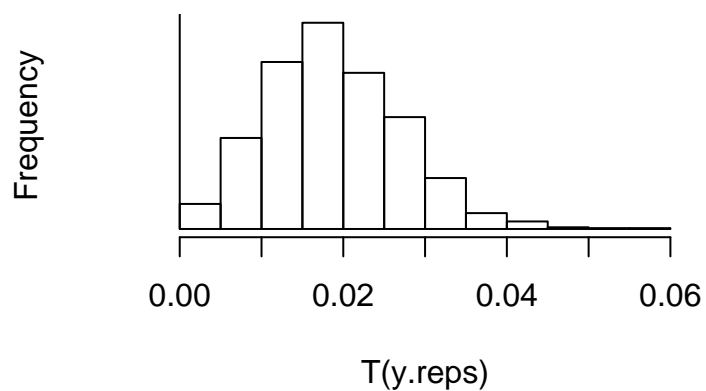
Now that I have (somewhat) recreated the assumed model fitted to the rat tumor data, I will define some test quantities of interest. I will test the maximum predicted proportion, the minimum predicted proportion, and the variability across all proportions. Although I am unsure these truly have “scientific interest,” I can imagine they might be of use for model checking (and they will provide me worthwhile practice in posterior predictive checking). Also, the text notes that the “posterior variability is higher in the full Bayesian analysis,” so I wanted to see if my model followed this assertion.

While the model does a poor job representing accurate minimum proportions, it is able to capture the maximum value quite well. As predicted in the text, the model produces much higher posterior variances than observed (the line representing the observed value isn’t even displayed on the histogram). P-values are also displayed in a table below the figures for reference.

Rat Tumor Model: Maximum Value



Rat Tumor Model: Minimum Value



Rat Tumor Model: Variance

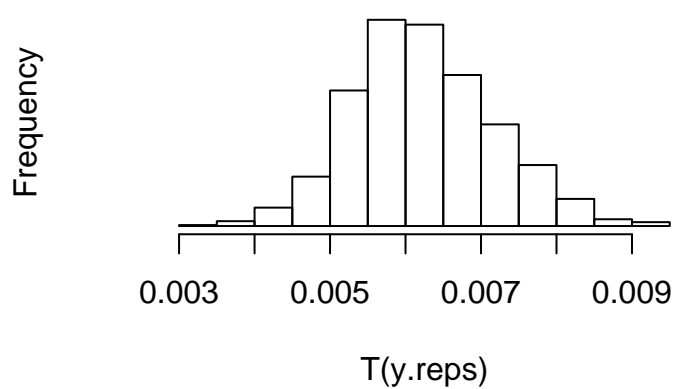


Table 6: P-Value Summary Radon Data

	Maximum	Minimum	Variance
P-Value	0.33	1	0

Chapter 7 Problem 5

7.5 Part A

As they direct in the problem statement, let us say that $p(\mu, \log \sigma, \phi) \propto \frac{1}{\sigma^2} p(\phi)$ where $p(\phi) \propto 1$ (i.e. uniform). Note that we retain the $\frac{1}{\sigma^2}$ because it translates to the uniform prior on $\log(\sigma)$. Our posterior would be:

$$p(\mu, \log \sigma, \phi | Y) \propto p(Y | \mu, \log \sigma, \phi) p(\mu, \log \sigma, \phi)$$

$$p(\mu, \log \sigma, \phi | Y) \propto \prod_i \sigma^{-1} \exp \left[\frac{-1}{2\sigma^2} \left(\frac{y_i^\phi - 1}{\phi} - \mu \right)^2 \right] * p(\phi)$$

We will use their hint and consider what happens when all data points y_i are multiplied by a constant factor. We will call this constant, $c \in \mathbb{R}$. We can use the variable transformation technique we learned in class:

$$p_v(v) = |J| p_u(f^{-1}(v))$$

Where $p_u(y_i) = \frac{y_i^\phi - 1}{\phi}$ and $f(v) = \frac{c^\phi y_i^\phi - 1}{\phi}$. Therefore,

$$p_v(v) = \left| \frac{du}{dv} \right| p_u \left(\frac{\phi y_i + 1}{c^\phi} \right)$$

$$p_v(v) = \left| \frac{1}{c^\phi} \right| \left(\frac{\frac{\phi y_i + 1}{c^\phi} - 1}{\phi} \right)$$

$$p_v(v) = \left(\frac{1}{c^\phi} \right) \left(\frac{\phi y_i + 1 - c^\phi}{\phi c^\phi} \right)$$

Because the Jacobian produces a factor of $\frac{1}{c^\phi}$ and the prior does nothing to account for it, any real number chosen for c will scale the data and will affect the mean and the variance.

7.5 Part B

Now let us set our prior to $p(\mu, \log \sigma, \phi) \propto \dot{y}^{1-\phi} p(\phi)$ as directed. Note that $\dot{y} = (\prod_{i=1}^n y_i)^{1/n}$. Now, our posterior takes the form:

$$p(\mu, \log \sigma, \phi | Y) \propto p(Y^\phi | \mu, \log \sigma, \phi) p(\mu, \log \sigma, \phi)$$

$$p(\mu, \log \sigma, \phi | Y) \propto \prod_i \sigma^{-1} \exp \left[\frac{-1}{2\sigma^2} \left(\frac{y_i^\phi - 1}{\phi} - \mu \right)^2 \right] * y_i^{\frac{1-\phi}{n}} p(\phi)$$

Now let us transform by a constant multiple. Our Jacobian remains unchanged from Part A and produces $\frac{1}{c^\phi}$. Therefore we can write,

$$p(\mu, \log \sigma, \phi | Y) \propto p(Y_c^\phi | \mu, \log \sigma, \phi) p(\mu, \log \sigma, \phi)$$

$$p(\mu, \log \sigma, \phi | Y) \propto \prod_i \sigma^{-1} \exp \left[\frac{-1}{2\sigma^2} \left(\frac{c^\phi y_i^\phi - 1}{\phi} - \mu \right)^2 \right] * \left(\frac{1}{c^\phi} \right) (c y_i)^{\frac{1-\phi}{n}} p(\phi)$$

If we investigate the final portion of the product:

$$\prod_i \left(\frac{1}{c^\phi} \right) (c y_i)^{\frac{1-\phi}{n}} = \left(\frac{1}{c^\phi} \right) \prod_i c^{\frac{1-\phi}{n}} y_i^{\frac{1-\phi}{n}}$$

$$\begin{aligned}
&= \left(\frac{1}{c^\phi} \right) \left(c^{\frac{1-\phi}{n}} \right)^n \prod_i y_i^{\frac{1-\phi}{n}} \\
&= \left(\frac{1}{c^\phi} \right) (c^{1-\phi}) \dot{y}^{1-\phi} \\
&= c \dot{y}^{1-\phi}
\end{aligned}$$

We are left with a only a constant that is not dependent on ϕ . We have shown that this prior accounts for any dependency on ϕ .

7.5 Part C

Now we wish to write the marginal posterior density $p(\phi|y)$ for the model derived in Part B. From our work above we have the form of $p(\mu, \sigma, \phi|y)$ and we wish to find the marginal: $p(\phi|y)$. To do so we must integrate the joint posterior over μ and σ :

$$\begin{aligned}
p(\phi|y) &= \int \int p(\mu, \sigma, \phi|y) d\mu d\sigma \\
p(\phi|y) &\propto \int \int p(y|\mu, \sigma, \phi) p(\mu, \sigma, \phi) d\mu d\sigma
\end{aligned}$$

Because the prior distribution: $p(\mu, \sigma, \phi)$ does not depend on μ or σ , we can pull it to the outside of the integral. Recall $p(\mu, \sigma, \phi) \propto \left(\prod_{i=1}^n y_i \right)^{\frac{1-\phi}{n}} p(\phi)$. Therefore we can write:

$$\begin{aligned}
p(\phi|y) &\propto p(\phi) \left(\prod_{i=1}^n y_i \right)^{\frac{1-\phi}{n}} \int \int p(y|\mu, \sigma, \phi) d\mu d\sigma \\
p(\phi|y) &\propto p(\phi) \left(\prod_{i=1}^n y_i \right)^{\frac{1-\phi}{n}} \int \int \prod_{i=1}^n \frac{1}{\sigma \sqrt{2\pi}} \exp \left[-\frac{1}{2} \left(\frac{y_i^{(\phi)} - \mu}{\sigma} \right)^2 \right] d\mu d\sigma \\
p(\phi|y) &\propto p(\phi) \left(\prod_{i=1}^n y_i \right)^{\frac{1-\phi}{n}} \int \int (2\sigma^2\pi)^{-\frac{n}{2}} \exp \left[-\frac{1}{2\sigma^2} \sum_{i=1}^n \left(y_i^{(\phi)} - \mu \right)^2 \right] d\mu d\sigma
\end{aligned}$$

I unfortunately ran out of time to solve this all the way out. I hope that my set up displays some of my basic knowledge of what I would do.

7.5 Part D

The prior distribution in (b) depends on the data, i.e. is outcome dependent (considered a non-Bayesian property). In general, we do not want to use the same data to formulate the prior because it usually plays the role of updating the prior to the posterior. Generally, the prior should be based on a priori knowledge (exclusive to the data). This could result in, for example, a much narrower posterior than is reflective of the true state of the measure.

7.5 Part E

The implicit truncation affects the model by making it only applicable in certain instances. It allows for larger variances, but creates a different relationship between mean and mode.

Chapter 7 Problem 6

7.6 Part A

I used Roe's code to fit the power-transformed normal model to the basement measurements of Blue Earth County. I chose a ϕ of 0.03125 to best suit the data.

```
y.blue.earth <- c(5.0, 13.0, 7.2, 6.8, 12.8, 9.5, 6.0, 3.8, 1.8, 6.9, 4.7, 9.5)
y.clay <- c(12.9, 2.6, 26.6, 1.5, 13.0, 8.8, 19.5, 9.0, 13.1, 3.6)
y.goodhue <- c(14.3, 7.6, 2.6, 43.5, 4.9, 3.5, 4.8, 5.6, 3.5, 3.9, 6.7)
y.all <- c(y.blue.earth,y.clay,y.goodhue)

#from roe's code
calcDist = function(lambda, x) {
  geoMean = exp(mean(log(x)))
  transVal1 = (x^lambda - 1) / geoMean^(lambda - 1)
  transVal2 = geoMean * log(x)
  transVal = (lambda == 0) * transVal2 + (lambda != 0) * transVal1
  return(var(transVal))
}

#find best phi
checkValue = c(1,0.5,0.25,0.125,0.0625,0.03125,0,-0.03125,-0.0625,-0.125,-0.25,-0.5)
#apply(checkValue, calcDist, y.blue.earth)
#0.03125

#calculate posterior
sigmaS = (12-1) * var((y.blue.earth)^(0.03125)) / rchisq(10000,df=12-1)
muS = sqrt(sigmaS / 12) * rnorm(10000) + mean((y.blue.earth)^(0.03125))
generYmis = function(x) {
  return(rnorm(length(y.blue.earth), mean = x[1], sd = sqrt(x[2])))
}
fullPost = cbind(muS, sigmaS)

#sample from posterior
yMisSamp = (apply(fullPost,1, generYmis))^(1/0.03125)
```

7.6 Part B

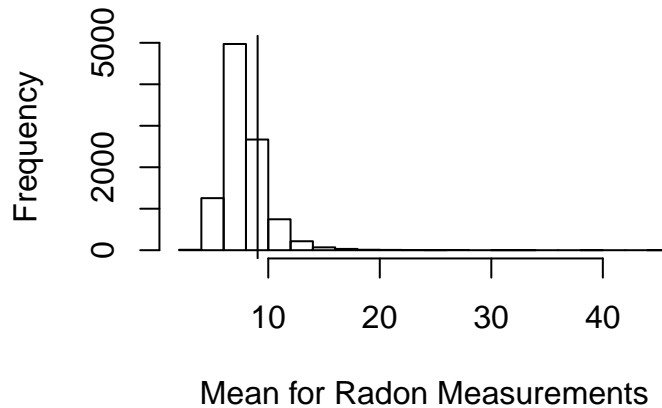
Now we wish to fit a power-transformed normal model to the basement measurements in all three counties. We are instructed to keep ϕ constant between them. I investigated and found that a ϕ of 0.03125 to best suited the data from Blue Earth and Clay. A ϕ of -0.03125 was slightly better for Goodhue, but only marginally. Therefore, I elected to use 0.03125.

I am a bit unclear about how best to “allowing the mean and variance of the normal distribution to vary” between the three counties. I had two thoughts. First, to simply build three normal distributions, one for each county, with the appropriate means and variances, using the same χ value. Second, to use a hierarchical model, as we did in the beginning of this problem set, to allow the means and variances of the counties to influence each other. I elected to use the first method for this report because (1) with our lack of apriori knowledge a conservative assumption might be that they should be treated separately and (2) because I was running out of time and I was struggling to get the second to work (see a draft of my attempt in the code appendix).

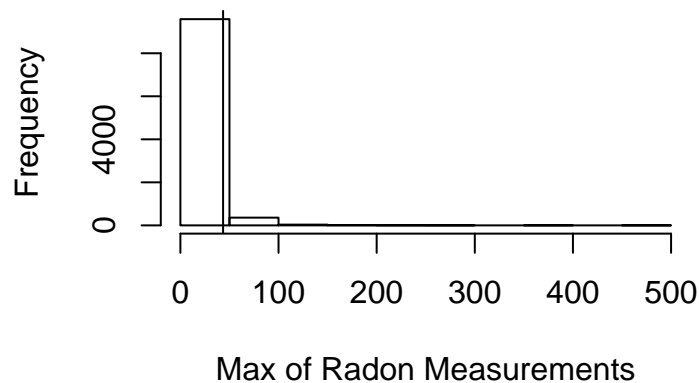
7.6 Part C

I checked the fit of my model using three posterior predictive simulations. I investigated the mean, the maximum, and the minimum value of the predicted radon measurements. See their plots and p-values below. The model did a relatively good job of representing all three test quantities. Although it often failed to predict maximum values above the true value, it was able to represent the minimum and mean values well. With more time I would be very interested in finishing and comparing my second suggested model's performance.

Test1: Mean for Radon Measurements



Test2: Max of Radon Measurements



Test3: Min of Radon Measurements

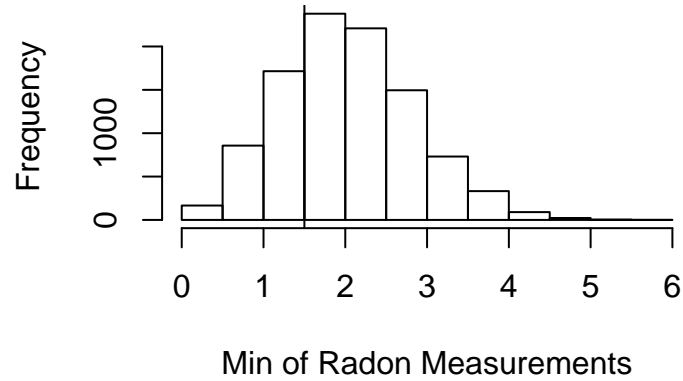


Table 7: P-Value Summary Radon Data

	Mean	Maximum	Minimum
P-Value	0.2008	0.0651	0.7262

7.6 Part D

It might be appropriate to fit a lognormal model to these data given that there are no negative values (also reason we might have been directed to use the power-transformed normal model). However, given that we choose a ϕ value that is not equal to 0, the lognormal model (employed if $\phi = 0$) would likely perform worse.

Code Appendix

```
knitr::opts_chunk$set(echo = F, results='asis', warning=F, message=F, cache=T)
library(kableExtra)
library(knitr)
#####
#####Heirarchical models
#####
library(lattice)

# output: one sample from p(theta | mu, tau, y)
conditional.theta=function(ybar,mu,tau,sigma){
  theta=rep(0,nschools)
  theta.hat=rep(0,nschools)
  V.hat=rep(0,nschools)
  for(j in 1:nschools) {
    V.hat[j]=1/(1/sigma[j]^2+1/(tau^2))
    theta.hat[j]=(ybar[j]/sigma[j]^2+mu/tau^2)*V.hat[j]
    theta[j]=rnorm(1,theta.hat[j],sqrt(V.hat[j]))
  }
  theta
}
```

```

}

# output: nsample samples from p(mu | tau, y)
sample.mar.mu=function(ybar, tau, sigma,nsample){
  V.mu.inv=sum(1/(sigma^2+tau^2))
  mu.hat=sum((1/(sigma^2+tau^2))*ybar)/V.mu.inv
  mu.sample=rnorm(nsample,mu.hat,sqrt(1/V.mu.inv))
  mu.sample
}

# evaluates p(tau | y)
marginal.tau=function(ybar,tau,sigma){
  V.mu.inv=sum(1/(sigma^2+tau^2))
  mu.hat=sum((1/(sigma^2+tau^2))*ybar)/V.mu.inv
  eval=exp(-(ybar-mu.hat)^2/(2*(sigma^2+tau^2)))
  eval=eval/sqrt(sigma^2+tau^2)
  eval=sqrt(1/V.mu.inv)*prod(eval)
  eval
}

ybar=c(28.39,7.94,-2.75,6.82,-0.64,0.63,18.01,12.16)
nschools=length(ybar)
sigma=c(14.9,10.2,16.3,11.0,9.4,11.4,10.4,17.6)
# Grid to evaluate p(tau | y)
x.tau=seq(0.00001,40,length=1000)
#x.tau=rep(Inf,length=1000)
#x.tau=rep(0,length=1000)

# evaluate p(tau | y) at 1000 points in the
#interval [0.00001,40]
post.tau=apply(t(x.tau),2,marginal.tau, ybar=ybar, sigma=sigma)
sample.tau=sample(x.tau,200,replace=TRUE,prob=post.tau)
# draw 200 samples from p(mu | tau, y)
sample.mu=apply(t(sample.tau),2,sample.mar.mu,ybar=ybar, sigma=sigma,nsample=1)
# draw 200 samples from p(theta | mu, tau, y)
sample.theta=matrix(0,ncol=nschools,nrow=200)
for (i in 1:200){
  sample.theta[i,]=conditional.theta(ybar, sample.mu[i], sample.tau[i],sigma)
}

# Expected posterior means E(theta_j | tau, y)
# averaging over mu
expected.theta=matrix(0,ncol=nschools,nrow=30)
x.tau.2=seq(0.00001,30,length=30)
for (i in 1:30){
  sample.mu=sample.mar.mu(ybar,x.tau.2[i],sigma, nsample=5000)
  sample.theta.2=matrix(0,ncol=nschools,nrow=5000)
  for (j in 1:5000){
    sample.theta.2[j,]=conditional.theta(ybar,sample.mu[j], x.tau.2[i],sigma)
  }
  expected.theta[i,]=apply(sample.theta.2,2,mean)
}

```



```

    expected.theta[i,]=apply(sample.theta.2,2,mean)
}

maxThetaJ = apply(sample.theta,1,max)
#hist(maxThetaJ,xlab="max(thetaj)")

#length(which(maxThetaJ > 28.4))/length(maxThetaJ)

## PART B
#prob best comp
Pr.best.matB <- matrix(NA, 8, 1)
for (i in 1:8) {
  Pr.best.matB[i] <- length(which(maxThetaJ == sample.theta[,i]))/length(maxThetaJ)
}
#prob better than others
Pr.comp.matB <- matrix(NA, 8, 8)
for (j in 1:8) {
  for (i in 1:8) {
    Pr.comp.matB[i,j] <- length(which(sample.theta[,i] > sample.theta[,j]))/length(sample.theta[,i])
  }
}
#combine into final dataframe
Pr.combB <- data.frame(Pr.best.matB, Pr.comp.matB)
colnames(Pr.combB) <- c("Pr(best)", "A", "B", "C", "D", "E", "F", "G", "H")
rownames(Pr.combB) <- c("A", "B", "C", "D", "E", "F", "G", "H")
#Pr.combB

#print table of data
kable(Pr.combB, "latex", caption = "Educational Testing Example w Tau --> Inf", booktabs = T) %>%
kable_styling(latex_options = c("striped", "hold_position"))
y.samples <- c(16/(16+58), 9/(9+90), 10/(10+48),
              13/(13+57), 19/(19+103), 20/(20+57),
              18/(18+86), 17/(17+112), 35/(35+273), 55/(55+64))
z.samples <- c(12/(12+113), 1/(1+18), 2/(2+14), 4/(4+44),
              9/(9+208), 7/(7+67), 9/(9+29), 8/(8+154))

y.values <- c(16, 9, 10, 13, 19, 20, 18, 17, 35, 55)
n.values <- c((16+58), (9+90), (10+48), (13+57), (19+103), (20+57),
              (18+86), (17+112), (35+273), (55+64))

library(lattice)
library(nleqslv)
nlocations<-length(y.values)

# first define hyperparameter prior
hyper.prior <- function(alpha,beta){
  alpha*beta*(alpha + beta)^(-5/2)
}

# next define marginal posterior
marginal.post <- function(alpha, beta) {
  post <- 1
  for (i in 1:nlocations) {

```

```

    if (n.values[i] > 100) n.values[i] = 100
    post <- post * (
      ((gamma(alpha + beta)) / (gamma(alpha) * gamma(beta))) *
      ((gamma(alpha + y.values[i]) * gamma(beta + n.values[i] - y.values[i])) /
      (gamma(alpha + beta + n.values[i]))))
    )
  }
  hyper.prior(alpha,beta) * post
}

#### build grid of alpha and beta values
#use nifty method of moments method like in PSET2
mom1 <- 1.356149
mom2 <- 8.615058

#build grid of transformed variables
trans.var1 <- seq(log(mom1/mom2)*1.5,log(mom1/mom2)/1.5,length.out=200)
trans.var2 <- seq(log(mom1+mom2)/3,log(mom1+mom2)*1.5,length.out=200)

#inverse to transform back to alpha and beta
beta <- exp(trans.var2)/(exp(trans.var1)+1)
alpha <- exp(trans.var2+trans.var1)/(exp(trans.var1)+1)

#calculate joint posterior density
post.dens <- outer(alpha,beta,function(x1,x2) log(marginal.post(x1, x2)))
post.dens <- exp(post.dens - max(post.dens))
post.dens <- post.dens/sum(post.dens)

#build contour plot of joint posterior
contours <- seq(min(post.dens), max(post.dens), length=10)
contour(trans.var1, trans.var2, post.dens,
  levels=contours,
  xlab="log(alpha/beta)",
  ylab="log(alpha+beta)",
  xlim=c(-2.2, -1.2),
  ylim=c(min(trans.var2), max(trans.var2)),
  drawlabels=FALSE,
  main="Contour plot of joint posterior")

#now draw samples from posterior
nsim <- 1000
dens.v1 <- apply(post.dens,1,sum)
tvar1.indices <- sample(1:length(trans.var1), nsim, replace=T, prob=dens.v1)
v1.samps <- trans.var1[tvar1.indices]
v2.samps <- rep(NA,nsim)
for (i in (1:nsim)) {
  v2.samps[i] <- sample(trans.var2, 1, prob=post.dens[tvar1.indices[i],])
}

plot(v1.samps, v2.samps,
  xlab="log(alpha/beta)^s",
  ylab="log(alpha+beta)^s",
  xlim=c(min(v1.samps), max(v1.samps)) ,
  ylim=c(min(v2.samps), max(v2.samps)),

```

```

    main="Sample Draws")

#inverse transform to get original hyperparameters
beta.samps <- exp(v2.samps) / (exp(v1.samps)+1)
alpha.samps <- exp(v2.samps+v1.samps) / (exp(v1.samps)+1)
theta.pull <- sapply(1:10, function(i)
  rbeta(1000, alpha.samps + y.values[i], beta.samps + n.values[i] - y.values[i]))
theta.sorted <- apply(theta.pull,2,sort)
plot(0:600/1000, 0:600/1000,
     type="l",
     xlab="Observed rate (y/n)",
     ylab="Median of posterior dist & 95% CI")
points(y.values/n.values,theta.sorted[500,])
segments(y.values/n.values,theta.sorted[25,], y.values/n.values,theta.sorted[975,] )
title(main="Posterior distribution of biker proportion")
y.finals <- alpha.samps/(alpha.samps+beta.samps)
y.finals.df <- data.frame(t(quantile(y.finals, c(0.025, 0.975))))
colnames(y.finals.df) <- c("2.5%", "97.5%")
rownames(y.finals.df) <- c("")

kable(y.finals.df, "latex", caption = "5.13 Part D", booktabs = T) %>%
kable_styling(latex_options = c("striped", "hold_position"))
theta.new <- rbeta(1000, alpha.samps, beta.samps)
y.new <- rbinom(1000, 100, theta.new)
#quantile(y.new, c(0.025, 0.975))
y.new.df <- data.frame(t(round(quantile(y.new, c(0.025, 0.975)),1)))
colnames(y.new.df) <- c("2.5%", "97.5%")
rownames(y.new.df) <- c("")

kable(y.new.df, "latex", caption = "5.13 Part E", booktabs = T) %>%
kable_styling(latex_options = c("striped", "hold_position"))
#load in data
airplane.data <- data.frame(c(1976:1985),
                             c(24,25,31,31,22,21,26,20,16,22),
                             c(734,516,754,877,814,362,764,809,223,1066),
                             c(0.19,0.12,0.15,0.16,0.14,0.06,0.13,0.13,0.03,0.15))
colnames(airplane.data) <- c("year", "f.accidents", "p.deaths", "d.rate")
n<-dim(airplane.data)[1]
airplane.data$p.miles <- airplane.data$p.deaths*10^8/(airplane.data$d.rate)
## first model, no of fatal accidents
### assumption 1: independent Poisson distributions
expected <- rep(sum(airplane.data$f.accidents)/n, 10)
test.stat <- rep(NA, 1000)
for (ii in 1:1000) {
  observed <- rep(NA, 10)
  for (i in 1:10){
    theta <- rgamma(1,sum(airplane.data$f.accidents))/n
    observed[i] <- rpois(1,theta)
  }
  test.stat[ii] <- sum((observed - expected)^2/expected)
}

test.stat.real <- sum((airplane.data$f.accidents - expected)^2/expected)

```



```

hist(test.stat, xlab="T(y.reps)", main="Model 1: Assumption 1", yaxt="n", cex=2)
pval1a <- length(which(test.stat > test.stat.real))/length(test.stat)
#pval1a
lines(rep(test.stat.real,2), c(0,1000))

### assumption 2: no trend over time
test.stat <- rep(NA, 1000)
for (ii in 1:1000) {
  observed <- rep(NA, 10)
  for (i in 1:10){
    theta <- rgamma(1,sum(airplane.data$f.accidents))/n
    observed[i] <- rpois(1,theta)
  }
  test.stat[ii] <- cor(observed,airplane.data$year)
}

test.stat.real <- cor(airplane.data$f.accidents,airplane.data$year)
hist(test.stat, xlab="T(y.reps)", main="Model 1: Assumption 2", yaxt="n", cex=2)
pval1b <- length(which(test.stat > test.stat.real))/length(test.stat)
lines(rep(test.stat.real,2), c(0,1000))
#pval1b
#plot(airplane.data$year,airplane.data$f.accidents)
#cor(airplane.data$year,airplane.data$f.accidents)
## second model: no of fatal accidents
### assumption 1: independent Poisson distributions proportional to number of miles flown
#expected <- rep(sum(airplane.data$f.accidents)/n, 10)
#expected changes, need to be proportional as well
expected <- (sum(airplane.data$f.accidents))*(airplane.data$p.miles/sum(airplane.data$p.miles))
test.stat <- rep(NA, 1000)
for (ii in 1:1000) {
  observed <- rep(NA, 10)
  for (i in 1:10){
    theta <- rgamma(1,sum(airplane.data$f.accidents))/sum(airplane.data$p.miles)
    observed[i] <- rpois(1,theta*airplane.data$p.miles[i])
  }
  test.stat[ii] <- sum((observed - expected)^2/expected)
}

test.stat.real <- sum((airplane.data$f.accidents - expected)^2/expected)
hist(test.stat, xlab="T(y.reps)", main="Model 2: Assumption 1", yaxt="n", cex=2)
pval2a <- length(which(test.stat > test.stat.real))/length(test.stat)
#pval2a
lines(rep(test.stat.real,2), c(0,1000))

### assumption 2: no trend over time
test.stat <- rep(NA, 1000)
for (ii in 1:1000) {
  observed <- rep(NA, 10)
  for (i in 1:10){
    theta <- rgamma(1,sum(airplane.data$f.accidents))/sum(airplane.data$p.miles)
    observed[i] <- rpois(1,theta*airplane.data$p.miles[i])
  }
}

```

```

test.stat[ii] <- cor(observed,airplane.data$year)
}

test.stat.real <- cor((sum(airplane.data$f.accidents))*(airplane.data$p.miles/sum(airplane.data$p.miles)
hist(test.stat, xlab="T(y.reps)", main="Model 2: Assumption 2", yaxt="n", cex=2)
pval2b <- length(which(test.stat > test.stat.real))/length(test.stat)
#pval2b
lines(rep(test.stat.real,2), c(0,1000))
#plot(sum(airplane.data$f.accidents)*(airplane.data$p.miles/sum(airplane.data$p.miles)),airplane.data$y
# third model: passenger deaths
### assumption 1: independent Poisson distributions
expected <- rep(sum(airplane.data$p.deaths)/n, 10)
test.stat <- rep(NA, 1000)
for (ii in 1:1000) {
  observed <- rep(NA, 10)
  for (i in 1:10){
    theta <- rgamma(1,sum(airplane.data$p.deaths))/n
    observed[i] <- rpois(1,theta)
  }
  test.stat[ii] <- sum((observed - expected)^2/expected)
}

test.stat.real <- sum((airplane.data$p.deaths - expected)^2/expected)
hist(test.stat, xlab="T(y.reps)", main="Model 3: Assumption 1", yaxt="n", cex=2)
pval3a <- length(which(test.stat > test.stat.real))/length(test.stat)
#pval3a
lines(rep(0.03,2), c(0,1000))

### assumption 2: no trend over time
test.stat <- rep(NA, 1000)
for (ii in 1:1000) {
  observed <- rep(NA, 10)
  for (i in 1:10){
    theta <- rgamma(1,sum(airplane.data$p.deaths))/n
    observed[i] <- rpois(1,theta)
  }
  test.stat[ii] <- cor(observed,airplane.data$year)
}

test.stat.real <- cor(airplane.data$p.deaths,airplane.data$year)
hist(test.stat, xlab="T(y.reps)", main="Model 3: Assumption 2", yaxt="n", cex=2)
pval3b <- length(which(test.stat > test.stat.real))/length(test.stat)
lines(rep(test.stat.real,2), c(0,1000))
#pval3b
#plot(airplane.data$year,airplane.data$p.deaths)

# fourth model: passenger deaths proportional to number of miles flown
### assumption 1: independent Poisson distributions
#expected <- rep(sum(airplane.data$p.deaths)/n, 10)
expected <- (sum(airplane.data$p.deaths))*(airplane.data$p.miles/sum(airplane.data$p.miles))
test.stat <- rep(NA, 100)
for (ii in 1:100) {

```

```

observed <- rep(NA, 10)
for (i in 1:10){
  theta <- rgamma(1,sum(airplane.data$p.deaths))/sum(airplane.data$p.miles)
  observed[i] <- rpois(1,theta*airplane.data$p.miles[i])
}
test.stat[ii] <- sum((observed - expected)^2/expected)
}

test.stat.real <- sum((airplane.data$p.deaths - expected)^2/expected)
hist(test.stat, xlab="T(y.reps)", main="Model 4: Assumption 1", yaxt="n", cex=2)
pval4a <- length(which(test.stat > test.stat.real))/length(test.stat)
#pval4a
lines(rep(0.9605436,2), c(0,1000))

### assumption 2: no trend over time
test.stat <- rep(NA, 1000)
for (ii in 1:1000) {
  observed <- rep(NA, 10)
  for (i in 1:10){
    theta <- rgamma(1,sum(airplane.data$p.deaths))/sum(airplane.data$p.miles)
    observed[i] <- rpois(1,theta*airplane.data$p.miles[i])
  }
  test.stat[ii] <- cor(observed,airplane.data$year)
}

test.stat.real <- cor((sum(airplane.data$p.deaths))*(airplane.data$p.miles/sum(airplane.data$p.miles)),
hist(test.stat, xlab="T(y.reps)", main="Model 4: Assumption 2", yaxt="n", cex=2)
pval4b <- length(which(test.stat > test.stat.real))/length(test.stat)
#pval4b
lines(rep(test.stat.real,2), c(0,1000))
#plot(airplane.data$year,airplane.data$p.deaths)
#plot(airplane.data$year,airplane.data$p.miles)
pvalues.df <- data.frame(c(pval1a, pval1b),
                        c(pval2a, pval2b),
                        c(pval3a, pval3b),
                        c(pval4a, pval4b))
pvalues.df <- t(pvalues.df)
rownames(pvalues.df) <- c("Model 1", "Model 2", "Model 3", "Model 4")
colnames(pvalues.df) <- c("Assumption 1", "Assumption 2")

kable(pvalues.df, "latex", caption = "P-Value Assumptions Summary", booktabs = T) %>%
kable_styling(latex_options = c("striped", "hold_position"))
#example in textbook (original set up)
test.quant <- NULL
for (i in 1:1000){
  theta <- rbeta (1,8,14)
  y.reps <- rbinom (20,1,theta)
  n <- length(y.reps)
  test.quant <- c(test.quant,sum(y.reps[2:n] != y.reps[1:(n-1)]))}

hist(test.quant, xlab="T(y.reps)", main="Original Protocol", yaxt="n",
      breaks=seq(-.5, max(test.quant)+.5), cex=2)
#new set up

```

```

test.quant <- NULL
for (i in 1:1000){
  theta <- rbeta(1,8,14)
  y.reps <- rbinom(1,1,theta)
  while (sum(y.reps==0) < 13)
    y.reps <- c(y.reps, rbinom(1,1,theta))
  n <- length(y.reps)
  test.quant <- c(test.quant, sum(y.reps[2:n] != y.reps[1:(n-1)]))}

hist(test.quant, xlab="T(y.reps)", main="Updated Protocol", yaxt="n",
      breaks=seq(-.5, max(test.quant)+.5), cex=2)
#build dataset
props<-c(0/20, 0/20, 0/20, 0/19, 0/18, 0/18, 1/18, 1/18, 2/25, 2/20, 1/10, 5/49,
          3/20, 2/13, 9/48, 4/20, 10/48, 4/19, 6/23, 5/19, 6/22, 0/20, 0/20, 0/20,
          0/17, 1/20, 1/20, 2/24, 2/23, 2/20, 2/19, 5/46, 3/27, 10/50, 4/20, 4/20,
          4/19, 4/19, 5/22, 6/20, 6/20, 6/20, 0/20, 0/19, 1/20, 1/20, 2/20, 2/20,
          2/17, 7/49, 4/20, 4/20, 11/46, 12/49, 16/52, 15/47, 0/19, 0/19, 1/19, 1/19,
          2/20, 2/20, 7/47, 3/20, 4/20, 4/20, 5/20, 5/20, 15/46, 9/24)
y.values<-c(0, 0, 0, 0, 0, 0, 1, 1, 2, 2, 1, 5, 3, 2, 9, 4, 10, 4, 6, 5, 6, 0, 0, 0,
            0, 1, 1, 2, 2, 2, 2, 5, 3, 10, 4, 4, 4, 4, 5, 6, 6, 6, 0, 0, 1, 1, 2, 2,
            2, 7, 4, 4, 11, 12, 16, 15, 0, 0, 1, 1, 2, 2, 7, 3, 4, 4, 5, 5, 15, 9)
n.values<-c(20,20,20,19, 18, 18, 18, 18, 25, 20, 10, 49, 20, 13, 48, 20, 48, 19, 23, 19, 22,
            20, 20, 20, 17, 20, 20, 24, 23, 20, 19, 46, 27, 50, 20, 20, 19, 19, 22,
            20, 20, 20, 20, 19, 20, 20, 20, 20, 17, 49, 20, 20, 46, 49, 52, 47, 19,
            19, 19, 20, 20, 47, 20, 20, 20, 20, 20, 46, 24)
nlocations<-length(n.values)

# first define hyperparameter prior
hyper.prior <- function(alpha,beta){
  alpha*beta*(alpha + beta)^(-5/2)
}

# next define marginal posterior
marginal.post <- function(alpha, beta) {
  post <- 1
  for (i in 1:nlocations) {
    if (n.values[i] > 100) n.values[i] = 100
    post <- post * (
      ((gamma(alpha + beta)) / (gamma(alpha) * gamma(beta))) *
      ((gamma(alpha + y.values[i]) * gamma(beta + n.values[i] - y.values[i])) /
       (gamma(alpha + beta + n.values[i]))))
    )
  }
  hyper.prior(alpha,beta) * post
}

#build grid of transformed variables using numbers provided in text
trans.var1 <- seq(-2.3,-1.3,length.out=100)
trans.var2 <- seq(1,5,length.out=100)

alpha <- seq(1.4/4,1.4*4,length.out = 100)
beta <- seq(8.6/4,8.6*4,length.out = 100)

#calculate joint posterior density

```

```

post.dens <- outer(alpha,beta,function(x1,x2) log(marginal.post(x1, x2)))
post.dens <- exp(post.dens - max(post.dens))
post.dens <- post.dens/sum(post.dens)

#build contour plot of joint posterior
contours <- seq(0.05*post.dens[50,50], 0.95*post.dens[50,50], length=10)
contour(trans.var1, trans.var2, post.dens,
        levels=contours,
        xlab="log(alpha/beta)",
        ylab="log(alpha+beta)",
        xlim=c(min(trans.var1), max(trans.var1)) ,
        ylim=c(min(trans.var2), max(trans.var2)),
        drawlabels=FALSE,
        main="Contour plot of joint posterior")

#now draw samples from posterior
nsim <- 1000
dens.v1 <- apply(post.dens,1,sum)
tvar1.indices <- sample(1:length(trans.var1), nsim, replace=T, prob=dens.v1)
v1.samps <- trans.var1[tvar1.indices]
v2.samps <- rep(NA,nsim)
for (i in (1:nsim)) {
  v2.samps[i] <- sample(trans.var2, 1, prob=post.dens[tvar1.indices[i],])
}

plot(v1.samps, v2.samps,
     xlab="log(alpha/beta)^s",
     ylab="log(alpha+beta)^s",
     xlim=c(min(v1.samps), max(v1.samps)) ,
     ylim=c(min(v2.samps), max(v2.samps)),
     main="Sample Draws")

#inverse transform to get original hyperparameters
beta.samps <- exp(v2.samps) / (exp(v1.samps)+1)
alpha.samps <- exp(v2.samps+v1.samps) / (exp(v1.samps)+1)
theta.pull <- sapply(1:70, function(i)
  rbeta(1000, alpha.samps + y.values[i], beta.samps + n.values[i] - y.values[i]))
theta.sorted <- apply(theta.pull,2,sort)
plot(0:450/1000, 0:450/1000,
     type="l",
     xlab="Observed rate (y/n)",
     ylab="Median of posterior dist & 95% CI")
points(y.values/n.values,theta.sorted[500,])
segments(y.values/n.values,theta.sorted[25,], y.values/n.values,theta.sorted[975,] )
title(main="Posterior distribution of rat tumor proportion")
## test stat is maximum value
test.stat <- rep(NA, 1000)
for (ii in 1:1000) {
  observed <- rep(NA, 10)
  for (i in 1:70){
    observed[i] <- rbeta(1, alpha.samps + y.values[i], beta.samps + n.values[i] - y.values[i])
  }
  test.stat[ii] <- max(observed)
}

```

```

}

test.stat.real <- max(props)
hist(test.stat, xlab="T(y.reps)", main="Rat Tumor Model: Maximum Value", yaxt="n", cex=2)
pvalmax <- length(which(test.stat > test.stat.real))/length(test.stat)
#pvalmax
lines(rep(test.stat.real,2), c(0,1000))

## test stat is minimum value
test.stat <- rep(NA, 1000)
for (ii in 1:1000) {
  observed <- rep(NA, 10)
  for (i in 1:70){
    observed[i] <- rbeta(1, alpha.samps + y.values[i], beta.samps + n.values[i] - y.values[i])
  }
  test.stat[ii] <- min(observed)
}

test.stat.real <- min(props)
hist(test.stat, xlab="T(y.reps)", main="Rat Tumor Model: Minimum Value", yaxt="n", cex=2)
pvalmin <- length(which(test.stat > test.stat.real))/length(test.stat)
#pvalmin
lines(rep(test.stat.real,2), c(0,1000))

## test stat is avg variability
test.stat <- rep(NA, 1000)
for (ii in 1:1000) {
  observed <- rep(NA, 10)
  for (i in 1:70){
    observed[i] <- rbeta(1, alpha.samps + y.values[i], beta.samps + n.values[i] - y.values[i])
  }
  test.stat[ii] <- var(observed)
}

test.stat.real <- var(props)
hist(test.stat, xlab="T(y.reps)", main="Rat Tumor Model: Variance", yaxt="n", cex=2)
pvalvar <- length(which(test.stat > test.stat.real))/length(test.stat)
#pvalvar
lines(rep(0.0107,2), c(0,1000))

pvalues.df <- data.frame(c(pvalmax, pvalmin, pvalvar))
pvalues.df <- t(pvalues.df)
rownames(pvalues.df) <- c("P-Value")
colnames(pvalues.df) <- c("Maximum", "Minimum", "Variance")

kable(pvalues.df, "latex", caption = "P-Value Summary Radon Data", booktabs = T) %>%
kable_styling(latex_options = c("striped", "hold_position"))
y.blue.earth <- c(5.0, 13.0, 7.2, 6.8, 12.8, 9.5, 6.0, 3.8, 1.8, 6.9, 4.7, 9.5)
y.clay <- c(12.9, 2.6, 26.6, 1.5, 13.0, 8.8, 19.5, 9.0, 13.1, 3.6)
y.goodhue <- c(14.3, 7.6, 2.6, 43.5, 4.9, 3.5, 4.8, 5.6, 3.5, 3.9, 6.7)

```

```

y.all <- c(y.blue.earth,y.clay,y.goodhue)

#from roee's code
calcDist = function(lambda, x) {
  geoMean = exp(mean(log(x)))
  transVal1 = (x^lambda - 1) / geoMean^(lambda - 1)
  transVal2 = geoMean * log(x)
  transVal = (lambda == 0) * transVal2 + (lambda != 0) * transVal1
  return(var(transVal))
}

#find best phi
checkValue = c(1,0.5,0.25,0.125,0.0625,0.03125,0,-0.03125,-0.0625,-0.125,-0.25,-0.5)
#sapply(checkValue, calcDist, y.blue.earth)
#0.03125

#calculate posterior
sigmaS = (12-1) * var((y.blue.earth)^(0.03125)) / rchisq(10000,df=12-1)
muS = sqrt(sigmaS / 12) * rnorm(10000) + mean((y.blue.earth)^(0.03125))
generYmis = function(x) {
  return(rnorm(length(y.blue.earth), mean = x[1], sd = sqrt(x[2])))
}
fullPost = cbind(muS, sigmaS)

#sample from posterior
yMisSamp = (apply(fullPost,1, generYmis))^(1/0.03125)
#find best phi when considering all three counties
#sapply(checkValue, calcDist, y.blue.earth) #0.03125
#sapply(checkValue, calcDist, y.clay) #0.03125
#sapply(checkValue, calcDist, y.goodhue) #-0.03125

#calculate posterior
sigma1 = (12-1) * var((y.blue.earth)^(0.03125)) / rchisq(10000,df=12-1)
sigma2 = (10-1) * var((y.clay)^(0.03125)) / rchisq(10000,df=10-1)
sigma3 = (11-1) * var((y.goodhue)^(0.03125)) / rchisq(10000,df=11-1)

mu1 = sqrt(sigmaS / 12) * rnorm(10000) + mean((y.blue.earth)^(0.03125))
mu2 = sqrt(sigmaS / 10) * rnorm(10000) + mean((y.clay)^(0.03125))
mu3 = sqrt(sigmaS / 11) * rnorm(10000) + mean((y.goodhue)^(0.03125))

generYmis = function(x) {
  return(rnorm(11, mean = x[1], sd = sqrt(x[2])))
}
fullPost1 = cbind(mu1, sigma1)
fullPost2 = cbind(mu2, sigma2)
fullPost3 = cbind(mu3, sigma3)

#sample from posterior
yMisSamp1 = (apply(fullPost,1, generYmis))^(1/0.03125)
yMisSamp2 = (apply(fullPost,1, generYmis))^(1/0.03125)
yMisSamp3 = (apply(fullPost,1, generYmis))^(1/0.03125)

yMisSamp <- rbind(yMisSamp1,yMisSamp2,yMisSamp3)
#####

```

```
#####Heirarchical models#####
library(lattice)

# output: one sample from p(theta | mu, tau, y)
conditional.theta=function(ybar,mu,tau,sigma){
  theta=rep(0,nschools)
  theta.hat=rep(0,nschools)
  V.hat=rep(0,nschools)
  for(j in 1:nschools) {
    V.hat[j]=1/(1/sigma[j]^2+1/(tau^2))
    theta.hat[j]=(ybar[j]/sigma[j]^2+mu/tau^2)*V.hat[j]
    theta[j]=rnorm(1,theta.hat[j],sqrt(V.hat[j]))
  }
  theta
}

# output: nsample samples from p(mu | tau, y)
sample.mar.mu=function(ybar, tau, sigma,nsample){
  V.mu.inv=sum(1/(sigma^2+tau^2))
  mu.hat=sum((1/(sigma^2+tau^2))*ybar)/V.mu.inv
  mu.sample=rnorm(nsample,mu.hat,sqrt(1/V.mu.inv))
  mu.sample
}

# evaluates p(tau | y)
marginal.tau=function(ybar,tau,sigma){
  V.mu.inv=sum(1/(sigma^2+tau^2))
  mu.hat=sum((1/(sigma^2+tau^2))*ybar)/V.mu.inv
  eval=exp(-(ybar-mu.hat)^2/(2*(sigma^2+tau^2)))
  eval=eval/sqrt(sigma^2+tau^2)
  eval=sqrt(1/V.mu.inv)*prod(eval)
  eval
}

# radon data
ybar=c(mu1,mu2,mu3)
nschools=length(ybar)
sigma=c(sigma1,sigma2,sigma3)

# Grid to evaluate p(tau | y)
x.tau=seq(0.00001,40,length=1000)
#x.tau=rep(Inf,length=1000)
#x.tau=rep(0,length=1000)

# evaluate p(tau | y) at 1000 points in the
#interval [0.00001,40]
post.tau=apply(t(x.tau),2,marginal.tau, ybar=ybar, sigma=sigma)
sample.tau=sample(x.tau,200,replace=TRUE,prob=post.tau)
# draw 200 samples from p(mu | tau, y)
sample.mu=apply(t(sample.tau),2,sample.mar.mu,ybar=ybar, sigma=sigma,nsample=1)
# draw 200 samples from p(theta | mu, tau, y)
sample.theta=matrix(0,ncol=nschools,nrow=200)
for (i in 1:200){
```



```

    sample.theta[i,]=conditional.theta(ybar, sample.mu[i], sample.tau[i],sigma)
}

# Expected posterior means  $E(\theta_j \mid \tau, y)$ 
# averaging over  $\mu$ 
expected.theta=matrix(0,ncol=nschools,nrow=30)
x.tau.2=seq(0.00001,30,length=30)
for (i in 1:30){
  sample.mu=sample.mar.mu(ybar,x.tau.2[i],sigma, nsample=5000)
  sample.theta.2=matrix(0,ncol=nschools,nrow=5000)
  for (j in 1:5000){
    sample.theta.2[j,]=conditional.theta(ybar,sample.mu[j], x.tau.2[i],sigma)
  }
  expected.theta[i,]=apply(sample.theta.2,2,mean)
}

#sample from posterior (original)
#fullPost = cbind(muS, sigmaS)
#yMisSamp = (apply(fullPost,1, generYmis))^(1/0.03125)

#posterior predictive checks

#on mean
yMisMeanCheck = apply(yMisSamp[1:33,],2,mean)
hist(yMisMeanCheck, xlab="Mean for Radon Measurements",main="Test1: Mean for Radon Measurements")
abline(v= mean(y.all))
pvalmean <- length(which(yMisMeanCheck > mean(y.all))) / length(yMisMeanCheck)
#pvalmean

#on maximum
yMisMaxCheck = apply(yMisSamp[1:33,],2,max)
hist( yMisMaxCheck, xlab="Max of Radon Measurements",main="Test2: Max of Radon Measurements")
abline(v= max(y.all))
pvalmax <- length(which(yMisMaxCheck > max(y.all))) / length(yMisMaxCheck)
#pvalmax

#on minimum
yMisMinCheck = apply(yMisSamp[1:33,],2,min)
hist( yMisMinCheck, xlab="Min of Radon Measurements",main="Test3: Min of Radon Measurements")
abline(v= min(y.all))
pvalmin <- length(which(yMisMinCheck > min(y.all))) / length(yMisMinCheck)
#pvalmin

pvalues.df <- data.frame(c(pvalmean, pvalmax, pvalmin))
pvalues.df <- t(pvalues.df)
rownames(pvalues.df) <- c("P-Value")
colnames(pvalues.df) <- c("Mean", "Maximum", "Minimum")

kable(pvalues.df, "latex", caption = "P-Value Summary Radon Data", booktabs = T) %>%
kable_styling(latex_options = c("striped", "hold_position"))

```