MAS8405 Bayesian data analysis: Project

B6030326 Callum Simpson

Introduction

In this report I will discuss my Bayesian investigation into the Reisby dataset to see how the antidepressant drug Imipramine effects the level of depression (recorded using the Hamilton depression index) in patients over several weeks.

Data

The hd score will be the prediction variable with the other features being the exploratory variables. Initially models where tested using unstandardized and standardized data however this report will be discussing the results of the models, I ran using standardized data. Data was standardized to help speed up model creation and improve mixing. I standardize the continuous explanatory variables but decided to keep binary variables as is. The hd score was kept as is to make the interpretation of results easier. See appendix 1. This data will be the “full” data. Later I will introduce “clean”.

Diagnostics

For Diagnostics I will be using 4 factors to determine how well my model has done. The first will be the thinning of trace plots to see how well the chains have mixed. This will backed up by checking the Gelman-Rubin statistics to see if all parameters equaled one suggesting that the chains have converged. I will also check the autocorrelation graphs of the thinned MCMC outputs to see if autocorrelation has been removed, this will then be backed up by a check of effective sample size to see if they are close to the best possible effective sample size, with the smaller the difference between the two the better the removal of auto correlation.

Prior distributions

For prior distribution I decided that I should make them as uninformative as possible. As this is a set of data that is new to me on a topic that I have no real world understanding of the work that goes on behind the scenes I assumed the effects of trying to manually assign distributions could lead to errors. Therefore, prior distribution will be normal distribution around mean zero dnorm(0, 1E-6) and variance will be broad dgamma (0.001,0.001) distribution. I will be discussing standard deviation over tau as it easier to interpret.

Other types of prior distribution where considered, tests with variance being calculated with Log normal and half-Cauchy prior were done however results there where pretty much the same as using dgamma. So, In the end I decided to stick with dgamma variance as I have more experience in using it.

Multiple Linear Regression

Multiple linear regression across the exploratory variables will be my first model. I felt like this model would act as a good start to my investigation as it’s a simple to use model and can be easily built upon. Multi Linear regression works by having one response variable y and a set of predictors x, and by using bayes we can work out the priors on each's exploratory variable to see how important that variable is to predicting y.

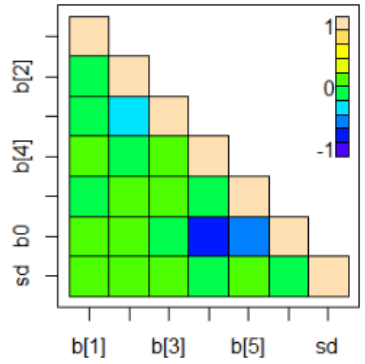
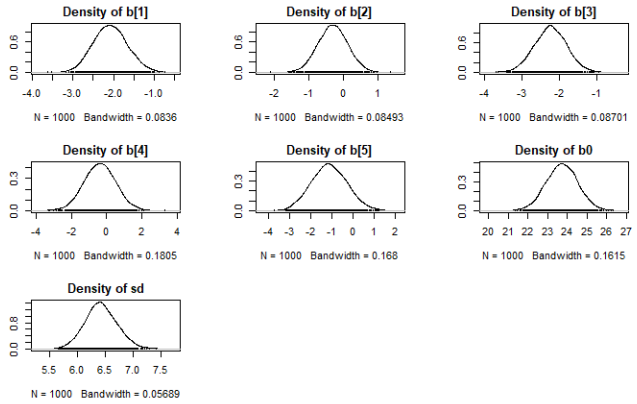
Multiple Linear Regression JAGS model
linReg_string = " model{

```
for (j in 1:p) {b[j]~dnorm(0, 1E-6)}

b0~dnorm(0, 1E-6)
tau ~ dgamma(0.001,0.001)
sd = pow(tau, -0.5)

for (i in 1:N){
   y[i,] ~ dnorm(mu[i],tau)
    mu[i] = b0 + inprod(b, x[i,])  }}"
```

Due to autocorrelation a thinning of 15 and burn in of at least 1500 was required to get the model ready (See Appendix 2). Checking on the diagnostics we see that the model has performed as desired. Looking at the trace plots we see the chains have converged well, this is backed up by the fact that Gelman-Rubin shows us that all parameters equal 1. When looking at the trace plots we see that there was no significant autocorrelation and the effective sample for all the parameters where all over 3900 out of 4000 suggesting that there is almost no correlation.

Looking at the correlation plot produce for the posterior distribution. We see that we have strong negative correlation between B[4] (female) and the intercept. We will later see why this is surprising. We also see that the intercept is also negatively correlated with the type of depression telling us that users depression has an effective on the starting depression intercept.

Looking at the density we see that the posterior distributions are approximately normal.

correlation plot	density
	

Multiple Linear Regression Parameter Estimate

Parameter	Mean	SD	High Interval (97.5%)	Low Interval (2.5%)
B[1] (week)	-2.0819	0.4135	-1.2711	-2.8922
B[2] (Lnimi)	-0.2943	0.4216	0.5336	-1.1190
B[3] (Lndmi)	-2.2482	0.4238	-1.4193	-3.0808
B[4] (female)	-0.3815	0.8996	1.4010	-2.145
B[5] (reactive_depression)	-1.1074	0.4125	0.5171	-2.751
B0	23.7549	0.8002	25.7617	22.1512
sd	6.4425	0.2927	7.0425	5.8951

When looking at the posterior means we gain a large understanding of how each variable effects the level of a person's depression. We see that the constant B0 has a mean of 23.75. We can take this as the base level of hd score before any of the other parameters are applied. This tells us that the base

level of depression in this data is moderately depressed. The 95% confidence interval is close to this mean, so it tells us that the assumption is accurate.

We see that there is a negative trend between b[1] (week) and level of Hamilton depression index. This tells us that as the week number increases the level of depression decreases. From this we can assume that the drug might be reducing a person's depression as the further they are into the tests the more of they have taken the drug. However, we can't say that this decrease in HD is 100% linked to the drug (external factors may be improving depression)

Next, I will discuss B[2] Lnimi and B[3] Lndmi. Lnimi is the concentration of the unmetabolized drug in the person blood stream and we see that it has very little effect on depression. This should be expected as if a raw drug hasn't been processed by the body, then it can't have an effect. Lndmi is the concentration of the drug in the patient's bloodstream that has been metabolized by the body. We see that the mean of B[3] is -2.25 with its credible interval's parameters being negative suggesting that the drug (when broken down by the body) does seem to have a significant effect on reducing a person depression.

B[4] is Gender (1 is female). We see that the mean is -0.38 which suggests the gender of the person does not seem to have much of an effect on the predicted HD. However, the 95% confidence interval is both strongly positive and negative hinting that a person's Gender might have an effect on the how well the drug might affect them. We also see that there is a stong correlation between Gender and B0 suggesting that the typical level of starting depression depends on the gender of the person.

Finally, b[5] the type of depression someone has seems to have some effect on how well the drug worked. If the mean was zero, we could say that the drug works on both types of depression equally, however as the 95% confidence interval goes above zero we cannot be sure as It tells us that reactive depression can cause an increase or decrease in HD levels.

The DIC Penalized deviance for this model was 1647. This will be the base model that I will build upon.

Non-linear Regression

Whilst we have some features that make sense to modeled as a linear function, I wanted to test if it would be best to model the Log concentration of IMI and DMI using a nonlinear regression. I believe that the breakdown of drugs isn't always linear and the metabolism of the drugs would slowly decrease over time. For my model I have included both linear and polynomial regression effect for the drug variables at the same time to see which is deemed more important. This model performed better than a model that only contained only polynomial regression.

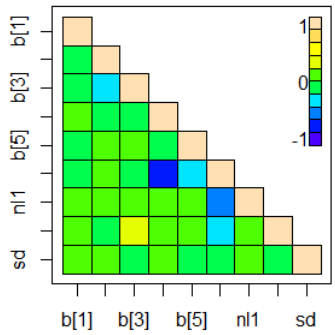
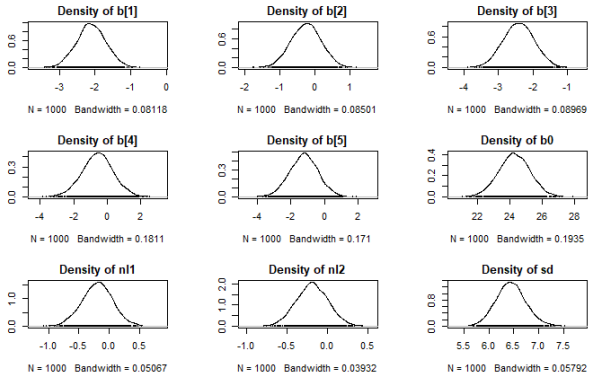
Non-linear Regression JAGS model

```
nl_linReg_string = " model{

b0~dnorm(0, 1E-6)
for (j in 1:p) {b[j]~dnorm(0, 1E-6)}
tau ~ dgamma(0.001,0.001)
sd = pow(tau, -0.5)

nl1 ~dnorm(0, 1E-6)
nl2 ~dnorm(0, 1E-6)

for (i in 1:N){
```

```
  y[i] ~ dnorm(mu[i],tau)
  mu[i] = b0 + b[1]*x[i,1] + b[2]*x[i,2] + nl1*pow(x[i,2],2) + b[3]*x[i,3] + nl2*pow(x[i,3],2) + b[4]*x[i,4] + b[5]*x[i,5]
}}"
```

This model needed a thinning of 20 with a burn in of 1000. See Appendix 3. Checking the diagnostics showed good performance, as there was no evidence of poor mixing in the trace plots of the Gelman Rubin diagnostic and the effective sample size was 3800 out of 4000 and there were no peaks in the auto correlation plot.

Looking at correlations we another strong negative correlation between Gender and b0. Looking at density we see that everything is normally distributed.

correlation plot	Density plots
	

Non-linear Regression Parameter Estimate				
Parameter	**Mean**	**SD**	**High Interval (97.5%)**	**Low Interval (2.5%)**
B[1] (week)	-2.0819	0.4135	-1.2711	-2.9388
B[2] (Lnimi linear)	-0.2943	0.4216	0.5336	-1.1190
B[3] (Lndmi linear)	-2.2482	0.4238	-1.4193	-3.0808
B[4] (female)	-0.5733	0.8996	1.4010	-2.145
B[5] (reactive_depression)	-1.1074	0.8351	0.5171	-2.751
B0	24.2901	0.4085	26.1772	22.4199
Nl1 (Lnimi non-linear)	-0.1920	0.2547	0.2821	-0.6884
Nl2 (Lndmi non-linear)	-0.1880	0.1974	0.1979	-0.5805
sd	6.4425	0.2927	7.0425	5.9040

We see that this model produced posterior means that are like our MLR model. Looking at nonlinear NL1 and Nl2 we see that these means are less than the means for the linear calculation of Lnimi and Lndmi. As these results are so close to zero, we can conclude that calculating using nonlinear has very little effect on the prediction of a user's depression.

In the end the DIC penalized deviance is 1650 which is slightly worse than the MLR model so it suggests that this model will not be effective in solving the problem.

Hierarchical Regression

Everyone is different. It's likely that the people in these trails have many different body types (body mass, health and metabolism rates) which would affect how the concentration of the drug effected a person. Because of this we should expect that some people may react differently to the drugs. Hierarchical Regression model could be used to solve this. We can group Users together by ID. This will

be useful as Hierarchical regression allows the sharing of data between groups by assuming that each group is sperate and everything in that group is like each other.

Hierarchical Regression JAGS model

```
Hie_Reg_string = "
model{

b0~dnorm(0, 1E-6)
for (j in 1:p) {b[j]~dnorm(0, 1E-6)}

tau ~ dgamma(0.001,0.001)
sd = pow(tau, -0.5)

tau_hie ~ dgamma(0.001,0.001)
sd_hie = pow(tau_hie, -0.5)

for (k in 1:subjects) {d[k]~dnorm(0, tau_hie)}

for (i in 1:N){
  y[i] ~ dnorm(mu[i],tau)
  mu[i] = b0 + inprod(b,x[i,]) + d[id[i]]
}}"
```

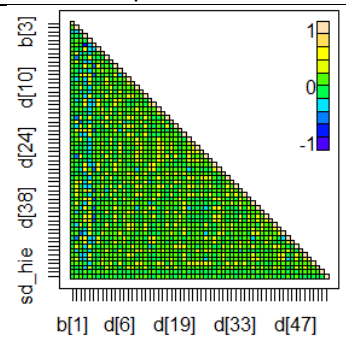
Due to the number of priors, we are producing (one group for each user) a lot of trails where needed to find the optimal amount of burning and thinning needed to remove the high levels of posterior correlation between each of the groups. After a lot of trials, it was found that large amount of thinning at 250 was required and a large burn in of 100000 was almost mandatory (appendix 4). Even though this model took a long time to run the diagnostics results shown that the removal of correlation and the merging of chains was effective.

Trace plots showed chain convergence with Gelman-Rubin shows us that all prior parameters are between 1 to 1.002 which is acceptable (trying to improve this is difficult). When looking at the trace plots we see that there was no significant autocorrelation and the effective sample for all the parameters where all over 3800 out of 4000 suggesting that there is almost no correlation.

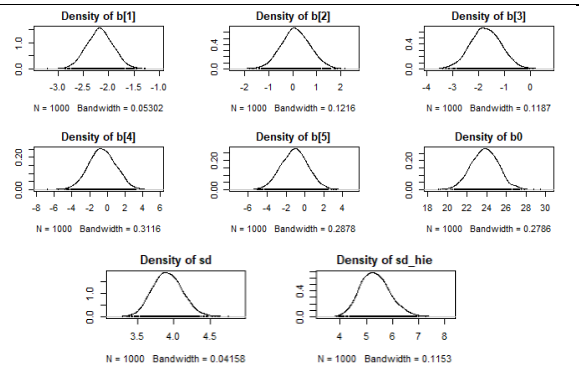
For density plots I will not be looking all the densities of every group as there would be too many to discuss. All results for our non-ID posterior distribution seems normal.

When looking at the pairs we see that there is very little correlation between the variables.

correlation plot



Density plots



Hierarchical Regression Parameter Estimate				
Parameter	**Mean**	**SD**	**High Interval (97.5%)**	**Low Interval (2.5%)**
B[1] (week)	-2.1679	0.2717	-1.6441	- 2.70115
B[2] (Lnimi)	0.11939	0.6780	1.0092	-1.67338
B[3] (Lndmi)	-1.71533	0.5884	-0.572902	-2.44385
B[4] (female)	-0.5922	1.7336	2.8201	-3.96811
B[5] (reactive_depression)	-1.08253	1.4264	1.742739	-3.83329
B0	23.4632	1.5841	26.6154	21.119684
sd	3.8221	0.2178	4.2680	3.42528
Sd_hier	5.1981	0.6288	6.5430	4.08178

We see that some of the trends that appeared in MLR seem to stay the same in this model. We see that a bit more importance has been placed on the gender of the person. This makes sense if we think that the build of person effects how effective the drug will be as males and females commonly have different builds. The idea that the groups of users can affect the usefulness of the drug is shown by the high sd_hier which is the prior variance for the group precision (users' difference from the average depression score). The fact that sd_hier is a high positive value and not close to zero suggests that there is some variation in the way that the drug affects patients differently. DIC is 1453.

Hierarchical Regression with Interaction Terms

We have two type of depression in this model, reactive depression and endogenous depression. We also know from the data explanation that antidepressants seem to have a stronger effect in those who have endogenous depression. I also wanted to try to see if there was any relation between a person's gender and the type of depression they have. This led me to try models using the Interactions between between Inimi ~ reactive depression, Indmi ~ reactive depression, females ~ reactive depression, Inimi ~ females, Indmi ~ females. Interactions are good as it allows us to make our model more complex by modifying our linear terms so we can see the effects of how one exploratory variable effect another.

The nice part about a lot of these models is that they build upon one base MLR model. Because of that I decided to try combine my previous well performing Hierarchical regression model with these interactions as the combination of beneficial features together could create a model that could be as optimized as possible. Originally, I did a MLR with Interaction terms and then this model however due to space and the results of the MLR with Interaction terms I just decide to talk about this model.

I tried all combinations of all the interactions and the following is the model that scored the best (DIC), tracking the relationship between Inimi ~ reactive depression and females ~ reactive depression.

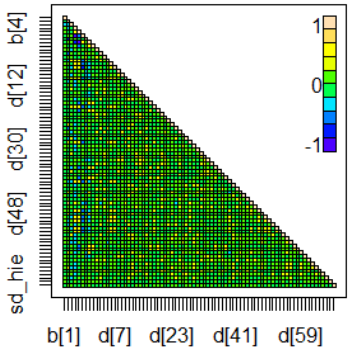
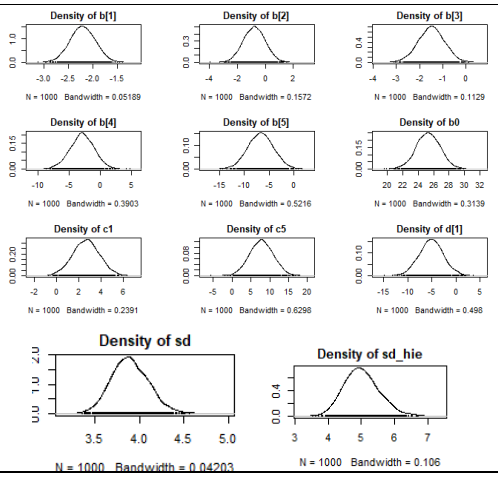
Combined Hierarchical Regression with Interaction Terms

```
mlr_combind_string = "
model{

b0~dnorm(0, 1E-6)
for (j in 1:p) {b[j]~dnorm(0, 1E-6)}

tau ~ dgamma(0.001,0.001)
sd = pow(tau, -0.5)

c1 ~ dnorm(0,1E-6)
c5 ~ dnorm(0,1E-6)

tau_hie ~ dgamma(0.001,0.001)
sd_hie = pow(tau_hie, -0.5)
```

```
for (k in 1:subjects) {d[k]~dnorm(0, tau_hie)}

for (i in 1:N){
  y[i] ~ dnorm(mu[i],tau)
  mu[i] = b0 + inprod(b,x[i,])  + c1*x[i,2]*x[i,5] + c5*x[i,4]*x[i,5] + d[id[i]]}}"
```

There were some issues with the c1 and c5 variables so a lot of burn in and thinning was needed. In the end I decided on a burn in of 100000 was needed with a thinning of 350. Like previous models, Diagnostics show both chains behaved well with as Trace plots and the Gelman-Rubin statistic showing no evidence of poor mixing (highest value is 1.002). Also, the effective sample size was over 3800 out of 4000. (Appendix 5)

Looking at the correlation plots we see a lot of correlation going on. The largest correlation is between the interaction terms and the two variables that make up that term which is what we should expect. Looking at the density plots we see that the distribution for each posterior is normal par from sd_hie which is slightly left skewed.

correlation plot	Density plots
	

Combined Hierarchical Regression with Interaction Terms Parameter Estimate				
Parameter	**Mean**	**SD**	**High Interval (97.5%)**	**Low Interval (2.5%)**
B[1] (week)	-2.21441	0.2539	- 1.72175	-2.7092
B[2] (Lnimi not reactive depression)	-0.75465	0.7883	0.76132	-2.3246
B[3] (Lndmi)	-1.48164	0.5565	-0.35740	-2.5668
B[4] (female not reactive depression)	-2.90715	1.9792	0.94096	-6.8755
B[5] (reactive depression)	-6.72547	2.6111	-1.62482	-11.9702
B0	25.33275	1.5515	28.49084	22.4242
c1 (Lnimi reactive depression)	2.59143	1.1637	4.95672	0.3373
c5 (female reactive depression)	7.86406	3.1169	14.03046	1.8429
sd	3.89085	0.2044	4.30532	3.5138
sd_hie	5.02817	0.5462	4.0481	6.18747

We see that this posterior summary is a lot different to the summaries of the other models we have created so far. First let's look at Lnimi and type of depression. We see that If the depression type is endogenous then it Lnimi causes a negative correlation (reduction in depression) where as if depression is reactive then we see that per lnimi unit cause a slight increase in a person depression. This helps

validate the theory that endogenous depression reacts well to anti-depressants. This might be because people who have reactive depression have difficulty breaking down the drug meaning they will have an excess of “raw” drug that’s not being used.

What we also see in our other relationship is that being female and having endogenous depression is negative correlated, again suggesting that the type of depression effects how well a person reacts to the treatment. This is backed up again by the fact females with reactive depression is highly positive again meaning that the type of depression can play an effect on how well a person reacts to the treatment/what there predicted depression will be.

Slightly off topic but, I found it surprising that relationships with Indmi frequently had a mean score close to zero. Lndmi seems to be a variable that is quite important in previous models so its relationship between it and other variables is not worth tracking. This suggests that no matter what the effectiveness of Lndmi and reducing depression isn't tied to anything. DIC is 1450.

Auto Regression model

One of the parameters we have got to work with is week. While we can use a predictor as a regression model sometimes we want to have a model that reacts to the previous time response. We have also seen in previous models that the week has a strong negative correlation with hd. We could assume that the higher the week number the longer the user has been on the antidepressant and therefore more chances for the drug to influence the user behavior. Because of this we may want to produce a model that utilizes time more. I decided that as a start I will use an AR(1) model. An issue is that the data is not a standard time series because we have multiple measurements in each week. To get around this I decided to clean the data (appendix 6) and then create a sort of hierarchal AR model that utilizes the HD value of the previous week for that user. Tests using higher AR chains where run but they got worse DIC values.

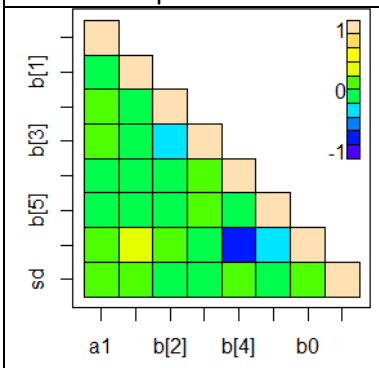
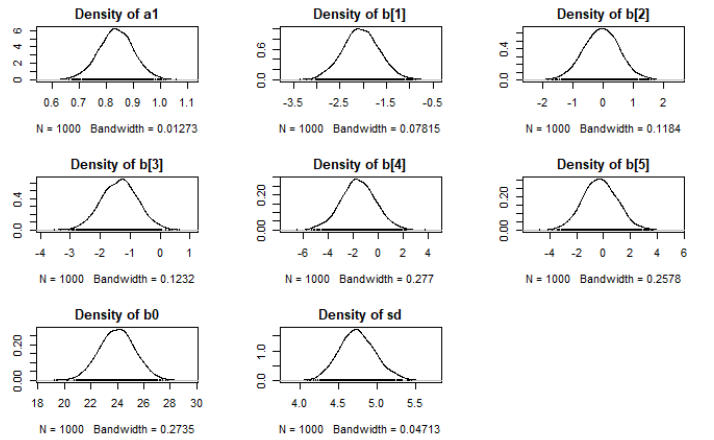
AR(1)

```
AR_1_string = "model{

b0~dnorm(0,  1E-6)
for (j in 1:p) {b[j]~dnorm(0, 1E-6)}

tau ~ dgamma(0.001,0.001)
sd = pow(tau, -0.5)
a1 ~ dnorm(0, 1E-6)

for (k in 1:users) {

  y[((k - 1) * 4) + 1] ~ dnorm(mu[((k - 1) * 4) + 1],tau)
  mu[((k - 1) * 4) + 1] = b0 + inprod(b,x[((k-1) * 4) + 1,])

  for (i in 2:4){
    y[((k - 1) * 4) + i] ~ dnorm(mu[((k - 1) * 4) + i] + a1 * (y[ ((k - 1) * 4) + i - 1] - mu[((k - 1) * 4) + i - 1]),tau)
   mu[((k - 1) * 4) + i] = b0 + inprod(b,x[((k-1) * 4) + i,]) }}}"
```

An issue with this model was that the amount of burn in needed was extremely high. A minimum burn in of 10000 was needed and a thinning of 125 was needed. Doing such extreme thinning paid off as Diagnostics show both chains mixed in the Trace plots and Gelman-Rubin statistic showed convergence. There was only one occasion in all our autocorrelation plots in which we had a bar go over 0.2 significant

correlation suggesting near all auto correlation was removed. Also, the effective sample size was over 3900 out of 4000. (Appendix 7)

We see that our posterior density seems to be pretty normal. Looking at the correlation plot produce We see that the plot produced is similar to the MLR model.

correlation plot	Density plots
	

AR(1) Parameter Estimate				
Parameter	**Mean**	**SD**	**High Interval (97.5%)**	**Low Interval (2.5%)**
a1	0.8068	0.08684	0.95419	0.704
B[1] (week)	-2.0571	0.38033	-1.31982	-2.8010
B[2] (Lnimi)	-0.06138	0.58688	1.11108	-1.219
B[3] (Lndmi)	-1.35118	0.61054	-0.1529	-2.549
B[4] (female)	-1.60085	1.40471	1.25634	-4.422
B[5] (reactive_depression)	-0.25554	1.27744	2.35659	-2.721
B0	23.98712	1.35530	26.62646	21.347
sd	4.75082	0.23457	5.23050	4.311

When compared to our original linear regression we see that there is a few changes in posterior means.

Firstly, Autoregression has put less emphasis on B[2] (Lnimi) putting its mean at near zero. This tells us that when taking the users pervious data, the level of un metabolized drug in the body in the grand scheme has no effect on if a person depression level. We also see that This model has put more emphasis on if a person gender. This AR model works similar to Hierarchal model (looping through groups of user's data) so it's possible using gender as a proxy to body type.

The main thing we want to look at is a1, this is the strength at which that the previous prediction of Hd and its mu help predict the next Hd value. An a1 of 1 is a random walk telling us the Auto regression in this model (0.806) might be a decaying random walk. So, the behavior of this model is to push any model that gets too large back down towards a point stopping the next predicted HR from increasing. This may be a bit farfetched but you can take this as the Autoregression pushing the HD prediction down over time, suggesting that the drug is lowering a person HD levels.

Auto Regression model with interaction terms.

I wanted to see If I could improve my AR model with the interaction's terms that were used above. I did do a test on a model that was the combination of auto regression, hierarchal regression and interaction terms however that model required a thinning of 1000 and gave back poor results.

Combined AR

```
AR_1_IT_string = "model{

b0~dnorm(0,  1E-6)
for (j in 1:p) {b[j]~dnorm(0, 1E-6)}

tau ~ dgamma(0.001,0.001)
sd = pow(tau, -0.5)

tau_hier ~ dgamma(0.001, 0.001)

a1 ~ dnorm(0, 1E-6)

c1 ~ dnorm(0,1E-6)
c5 ~ dnorm(0,1E-6)

for (k in 1:users) {
    y[((k - 1) * 4) + 1] ~ dnorm(mu[((k - 1) * 4) + 1],tau)

    mu[((k - 1) * 4) + 1] = b0 + inprod(b,x[((k-1) * 4) + 1,])  + c1*x[((k-1) * 4) + 1,2]*x[((k-1) * 4) + 1,5] + c5*x[((k-1) * 4) + 1,4]*x[((k-1) * 4) + 1,5]

  for (i in 2:4){
      y[((k - 1) * 4) + i] ~ dnorm(mu[((k - 1) * 4) + i] + a1 * (y[ ((k - 1) * 4) + i - 1] - mu[((k - 1) * 4) + i - 1]),tau)

     mu[((k - 1) * 4) + i] = b0 + inprod(b,x[((k-1) * 4) + i,])  + c1*x[((k-1) * 4) + i,2]*x[((k-1) * 4) + i,5] + c5*x[((k-1) * 4) + i,4]*x[((k-1) * 4) + i,5]}}}"
```

This model required a burn in of 100000 and a thinning of at least a thinning of 300. This produced well converged chains (good trace plots and Gelman-Rubin statistics to 1) and removed auto correlation (Actually effective size was 4000/4000). (Appendix 8)

The Density plots look normal and the correlation plots is similar to Linear Regression with Interaction Terms correlation plots. So, the largest correlation plot is between the interaction terms and the two variables that make up that term which is what we should expect.

Combined AR Parameter Estimate

Parameter	Mean	SD	High Interval (97.5%)	Low Interval (2.5%)
a1	0.8157	0.06856	0.9493	0.6810
B[1] (week)	-2.0869	0.38033	-1.3405	-2.8394
B[2] (Lnimi not reactive depression)	-1.1019	0.6593	0.4185	-2.6445
B[3] (Lndmi)	-1.2208	0.61054	0.0200	-2.4037
B[4] (female not reactive depression)	-1.9753	1.93563	1.91929	-5.6838
B[5] (reactive_depression)	-2.4580	2.49203	2.5032	-7.2056
B0	24.1933	1.60762	27.2397	21.0415
c1 (Lnimi reactive depression)	2.8087	1.13325	5.0183	0.5819
c5 (female reactive depression)	3.3050	2.99294	9.0757	-2.6362
sd	4.6853	0.23672	5.1856	4.2529

Looking at this model and the other model that included relationships we see a difference in the importance between the Parameter means of the interactions, telling us the A1 doesn't take these interactions to be that important. We see A1 is the same suggesting that other AR(1) models may also have a decaying random walk behavior

Final model selected

To compare my models, I ran a deviance information criterion simulation (dic.samples(...)) to get a value that I could use to choose which model was better/more appropriate than the other. I will be looking for a model with the lowest penalized deviance. However, we have a problem, two sets of models created on two different sets of data (full and clean). This is a problem as the difference between the data means we can't do a direct DIC comparison as the data is essentially on different scales. To get around this I decided to run all my models again but using the clean data (use appendix 6 but change the clean_x to stand_x and clean_y_hd to stand_y_hd) to get a new set of DIC so I could compare all my models together.

Model Clean	**Mean deviance**	**Penalty**	**Penalized deviance**
Multiple Linear Regression	1354	7.079	1361
Nonlinear Regression	1355	9.166	1364
Hierarchical Regression	1147	49.81	1197
AR	1238	8.13	1246
Hierarchical Regression with interaction Terms	1146	50.08	1196
AR with interaction terms	1233	10.14	1243

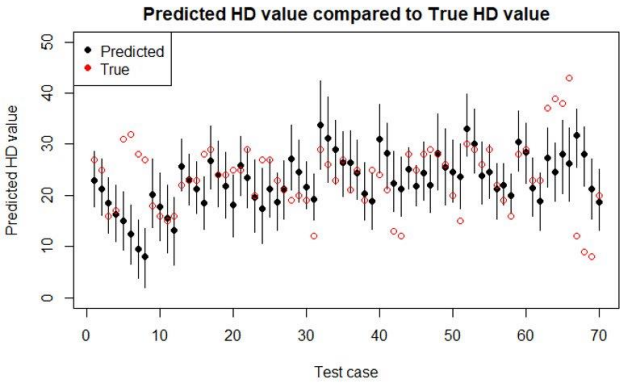
Looking at this the best model was Hierarchical Regression with interaction Terms with lowest DIC Penalized deviance at 1196. This is one of the models I ran with full data so I decided to run another DIC check with all the models that I made with the full data.

Model Full data	**Mean deviance**	**Penalty**	**Penalized deviance**
Multiple Linear Regression	1640	7.074	1647
Nonlinear Regression	1641	9.166	1650
Hierarchical Regression	1391	62.14	1453
Hierarchical Regression with interaction Terms	1389	61.64	1450

Looking at this the best model was Hierarchical Regression with interaction Terms again with a DIC Penalized deviance of 1450.

We see that in both cases Hierarchical Regression with interaction Terms is the best model and both reduce Penalized deviance by roughly 200. I will say that my final model will be the model that was made using Full data. My reasoning is that it will have been created using more data suggesting that its Parameter posterior may be more accurate and therefore better at predicting a user's depression level.

Finally, I wanted to test how effective my Hierarchical Regression with interaction Terms model was at predicting the HD whilst also looking at the actual values of HD



After writing the code in Appendix 9 I was able to create the graph above. We see that for a majority of the test cases that our model was able to catch the actual hd value within its 95% credible interval suggesting that in some cases the model is effective. However, we see If the HD value is extreme or follows an odd trend (like in the last user where there HD spikes in the last week) the model doesn't fit the data that well. This means our model has some difficulty with outliers. Suggesting our model could be improved and a method to try and deal with these fringe cases should be implement.

Conclusion

In conclusion, we have learned 3 important factors about the effectiveness of the antidepressant.

Firstly, just about every model has shown that Lndmi is negatively correlated with the hd depression score. This is important as Lndmi is the measure of “active” drug in a user's system so this negative correlation between the working drug and the depression score tells us that that this drug is working as intended and is combating depression. We could also sort of back this up with the fact that the week number tended to have a negative correlation as well, so the longer the user has been taking the drug then it's predicted that they will have a lower depression level.

Secondly, we see that the type of depression a person has does affect the actual effectiveness of the drug. We see large decrease in a user's depression if they endogenous depression quite quickly whereas we don’t see the same level of depression reduction in reactive depression. Telling us that the argument “Antidepressant medication is some-times argued to be more effective for endogenous depression” holds in this data. We see this in my final model where Lnimi not reactive depression is negative correlation but Lnimi and reactive depression is a strong positive correlation.

Finally, whilst we see the drug does seem to lower depression levels in all users, the rate and amount varies from person to person and the user themselves play a big part in determining how effective the drug is (this is backed up by the fact our best model told us the average users depression from the mean is around 5 units). This suggests that If we were to get more data about the users, then we would be able to build models that better fit the problem at hand. If I had time, I believe that checks gender could help verify this as I expect the build difference between men and women to play an impact in drug effectiveness

However, looking at my graph of using my model to predict a user depression level we see that it struggled in a lot of cases suggesting that we will need more data than the 66 users and a more complex model that better fits the data if we want to do any accurate depression score prediction.

APPENDIX

Appendix 1: Data load in , Full data

```
load("./Reisby.Rdata")

reisby = data.frame(Reisby)
ids = as.integer(as.factor(reisby[,1]))

remove_id = reisby[,-1]

for (i in 2:4) {
   z = remove_id[,i]
   remove_id[,i] = (z - mean(z))/sd(z)}

#remove_id
stand_y_hd = remove_id[,1]

stand_x = remove_id[,-1]
```

Appendix 2 : Multiple Linear Regression code to run sample

```
linReg_data =  list(y = stand_y_hd, x = stand_x , N = nrow(stand_x), p = ncol(stand_x))

linReg_model = jags.model(file = textConnection(linReg_string), data = linReg_data, n.chains=4)

update(linReg_model, n.iter=1500)

linReg_sample = coda.samples(model = linReg_model, variable.names =
c("b0","b","sd"),n.iter=15*1000, thin=15)
```

Appendix 3 : Nonlinear Regression code to run sample

```
nl_linReg_data =  list(y = stand_y_hd, x = stand_x , N = nrow(stand_x), p = ncol(stand_x))

nl_linReg_model = jags.model(file = textConnection(nl_linReg_string), data = nl_linReg_data,
n.chains=4)

##do 1000 iterations
update(nl_linReg_model, n.iter=1000)

## do 1000 iterations but save results
nl_linReg_sample = coda.samples(model = nl_linReg_model, variable.names =
c("b0","b","sd","nl1","nl2"),n.iter=20*10000, thin=20)
```

Appendix 4 : Code needed to run Hierarchical Regression

```
 Hie_Reg_data =  list(y = stand_y_hd, x = stand_x , p = ncol(stand_x), N = nrow(stand_x), id = ids
,subjects = length(unique(ids)))

Hie_Reg_model = jags.model(file = textConnection(Hie_Reg_string), data = Hie_Reg_data, n.chains=4)

update(Hie_Reg_model, n.iter=100000)

Hie_Reg_sample = coda.samples(model = Hie_Reg_model, variable.names =
c("b0","b","sd","sd_hie","d"),n.iter=250*1000, thin=250)
```

Appendix 5 : Code needed to run Hierarchical Regression with interaction Terms

```
mlr_combind_data =  list(y = stand_y_hd, x = stand_x , p = ncol(x), N = nrow(stand_x), id = ids
,subjects = length(unique(ids)))

mlr_combind_model = jags.model(file = textConnection(mlr_combind_string), data =
mlr_combind_data, n.chains=4)

update(mlr_combind_model, n.iter=100000)

mlr_combind_sample = coda.samples(model = mlr_combind_model, variable.names =
c("b0","b","sd","sd_hie","c1","c5","d"),n.iter=400*1000, thin=400)
```

Appendix 6 : Clean data for auto regression.

```
load("./Reisby.Rdata")

reisby = data.frame(Reisby)
ids = as.integer(as.factor(reisby$id))
id_frequencies = table(ids)
complete_ids = names(id_frequencies)[which(id_frequencies==4)]
complete_ids = as.numeric(complete_ids)
Reisby_complete = subset(Reisby, ids %in% complete_ids)

#Reisby_complete
ids2 = as.integer(as.factor(Reisby_complete[,1]))

remove_id = Reisby_complete[,-1]

for (i in 2:4) {
   z = remove_id[,i]
   remove_id[,i] = (z - mean(z))/sd(z)}


#head(remove_id)
clean_y_hd = remove_id[,1]
```

```
clean_x = remove_id[,-1]
```

Appendix 7 : Run Auto regression.

```
AR_1_data =  list(y = clean_y_hd, x = clean_x , N = nrow(clean_x), p = ncol(clean_x), id = ids2 ,users = length(unique(ids2)))

AR_1_model = jags.model(file = textConnection(AR_1_string), data = AR_1_data, n.chains=4)

update(AR_1_model, n.iter=10000)

AR_1_sample = coda.samples(model = AR_1_model, variable.names = c("b0","b","sd","sd_hier","a1"),n.iter=125*1000, thin=125)
```

Appendix 8: Run Auto regression with interaction terms

```
AR_1_IT_data =  list(y = clean_y_hd, x = clean_x , N = nrow(clean_x), p = ncol(clean_x), id = ids ,users = length(unique(ids)))

AR_1_IT_model = jags.model(file = textConnection(AR_1_IT_string), data = AR_1_IT_data, n.chains=4)

update(AR_1_IT_model, n.iter=100000)

AR_1_IT_sample = coda.samples(model = AR_1_IT_model, variable.names = c("b0","b","sd","sd_hier","c1","c5","a1"),n.iter=300*1000, thin=300)
```

Appendix 9: Hierarchical Regression prediction

```
xtrain = stand_x[1:180,]
xtest = stand_x[181:250,]

ytrain =  stand_y_hd[1:180]
ytest = stand_y_hd[181:250]

Hie_Reg_string = "
model{

b0~dnorm(0, 1E-6)
for (j in 1:p) {b[j]~dnorm(0, 1E-6)}

tau ~ dgamma(0.001,0.001)
sd = pow(tau, -0.5)

tau_hie ~ dgamma(0.001,0.001)
sd_hie = pow(tau_hie, -0.5)
```

```
c1 ~ dnorm(0,1E-6)
c5 ~ dnorm(0,1E-6)

for (k in 1:subjects) {d[k]~dnorm(0, tau_hie)}

for (i in 1:N){
  y[i] ~ dnorm(mu[i],tau)
  mu[i] = b0 + inprod(b,x[i,])   + c1*x[i,2]*x[i,5] + c5*x[i,4]*x[i,5] + d[id[i]]}

for (i in 1:nnew){
  y_pred[i]  = b0 + inprod(b,x_new[i,])  + c1*x_new[i,2]*x_new[i,5] + c5*x_new[i,4]*x_new[i,5] +
d[id[i]]

}}"


ID = as.integer(as.factor(reisby[,1]))

Hie_Reg_data =  list(y = ytrain, x = xtrain, nnew =  nrow(xtest),  x_new = xtest , p = ncol(x), N =
nrow(xtrain), id = ID ,subjects = length(unique(ID)))

Hie_Reg_model = jags.model(file = textConnection(Hie_Reg_string), data = Hie_Reg_data, n.chains=4)

update(Hie_Reg_model, n.iter=10000)

samples_pred = coda.samples(model= Hie_Reg_model, variable.names=c("y_pred"),n.iter=400*1000,
thin=400)

mean_pred = summary(samples_pred)$statistics[,1]

hdi_pred = HPDinterval(samples_pred[[1]])

plot(1: nrow(xtest), mean_pred,main="Predicted HD value compared to True HD value", xlab="Test
case",ylab="Predicted HD value", pch=16, ylim=c(0,50))
legend("topleft", legend=c("Predicted", "True"), col=1:2, pch=16)

for (i in 1: nrow(xtest)) {lines(x=rep(i, 2), y=hdi_pred[i,])}

for (i in 1: nrow(xtest)) {points(x=i, y=ytest[i], col = "red")}
```