Bayesain Project

Ammar Hasan

18 February 2019

Introduction

This report summaries the Bayesian analysis of the Reisby data-set. The Reisby data-set is based on a 5 week (first placebo) psychiatric study which investigates response of depressed patients to IMI. The Bayesian analysis will investigate how the drug affects depression.

Data Setup and Cleaning

Data

The data is loaded from a .Rdata file containing the Riesby data-set introduced in the introduction.

Data Exploration

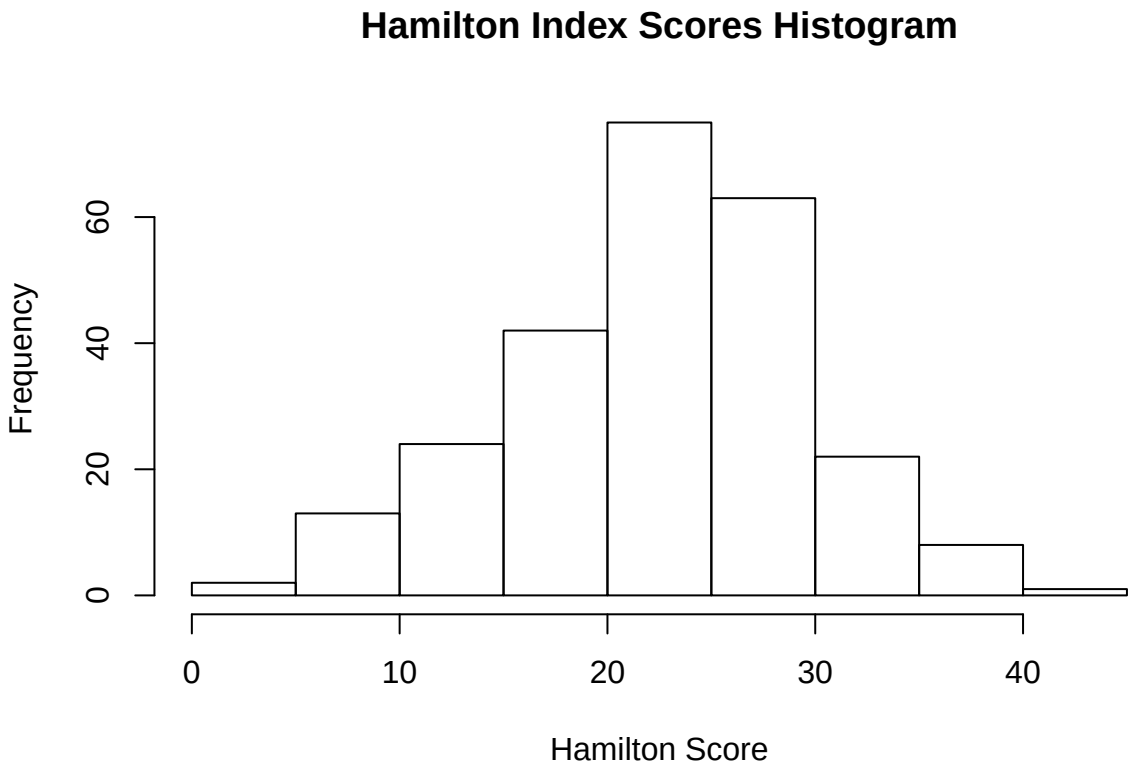
Correlation (Pairs Plot)

	hd	week	lnimi	lndmi	female	reactive_depression
hd	1.0000000	-0.3322674	-0.1165766	-0.3521130	-0.0624613	-0.0561811
week	-0.3322674	1.0000000	0.0357592	0.1239101	-0.0118326	0.0325663
lnimi	-0.1165766	0.0357592	1.0000000	0.2102979	0.0859277	-0.0365988
lndmi	-0.3521130	0.1239101	0.2102979	1.0000000	0.0945810	-0.1001591
female	-0.0624613	-0.0118326	0.0859277	0.0945810	1.0000000	0.1158473
reactive_depression	-0.0561811	0.0325663	-0.0365988	-0.1001591	0.1158473	1.0000000

The strongest correlations are negative and weak-moderate, and occur between Hamilton index with week and DMI, and rather obvious as the increase in blood concentration of the antidepressant would alleviate depression over weeks of treatment. Most other correlations are weak to very weak.

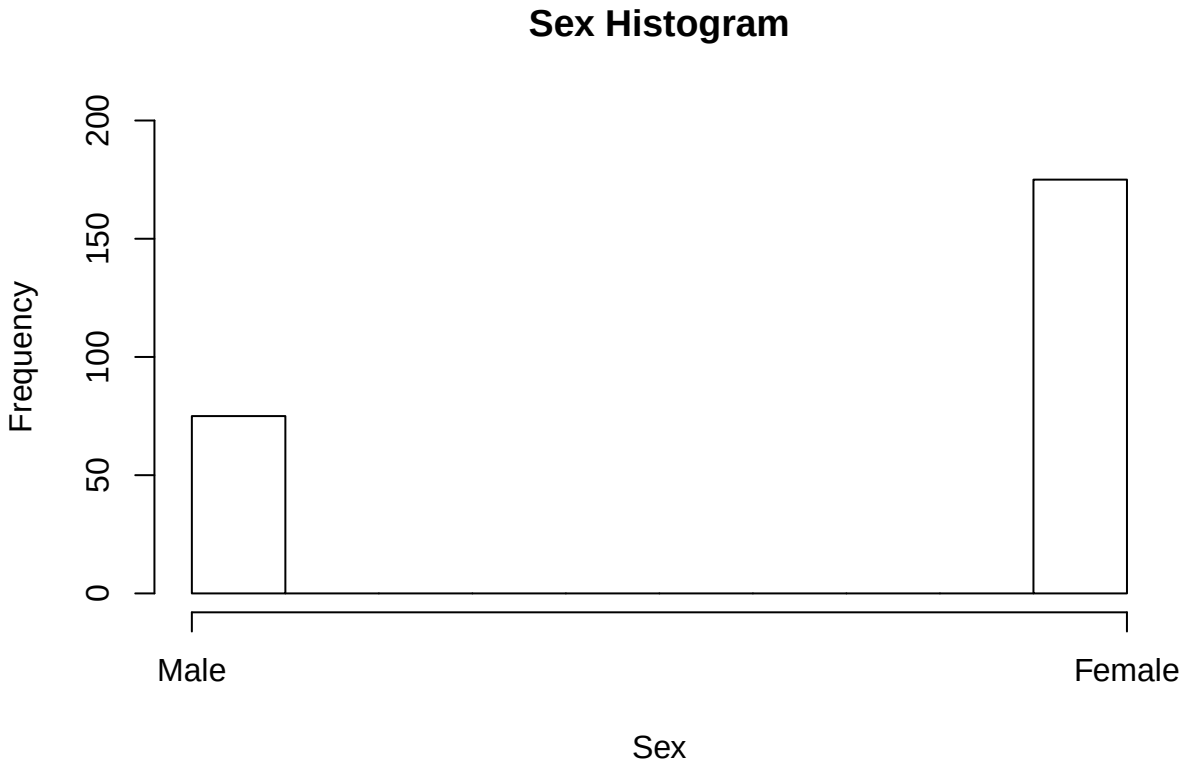
Graphical Summaries

Hamilton Scores



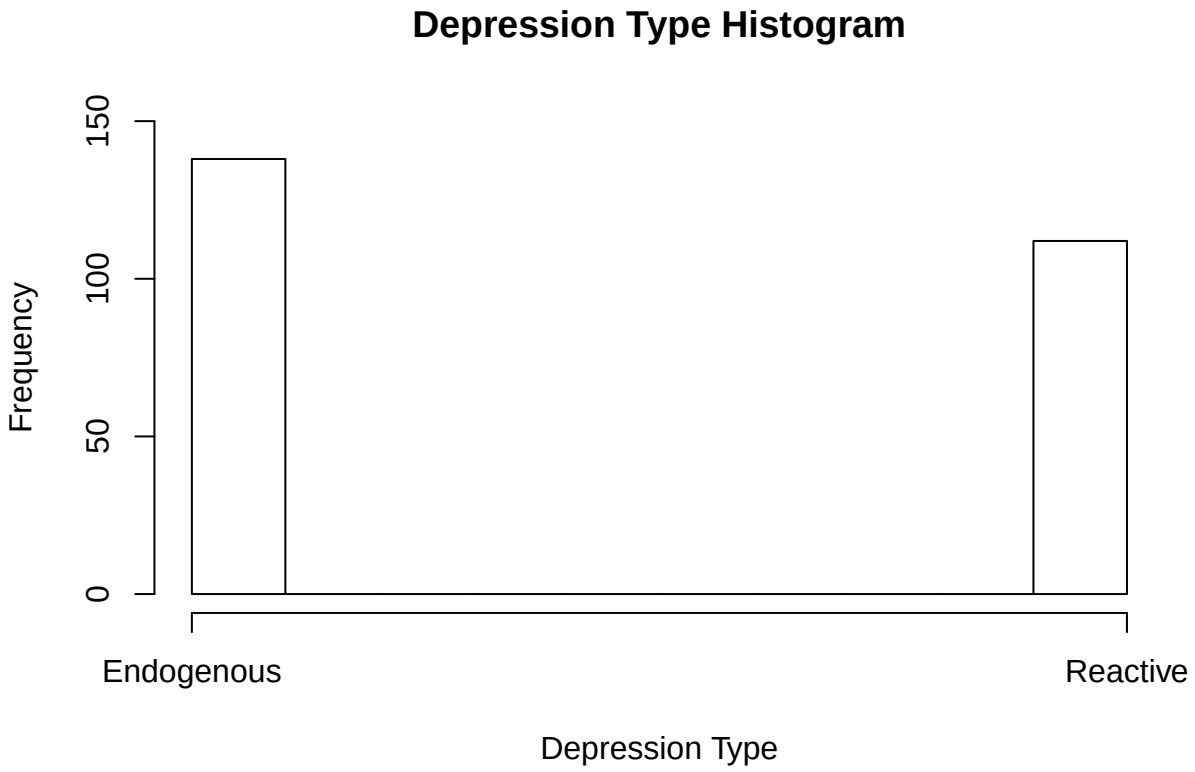
Most Hamilton scores are above 20 (i.e. moderate and severe depression dominates against mild and normal).

Sex



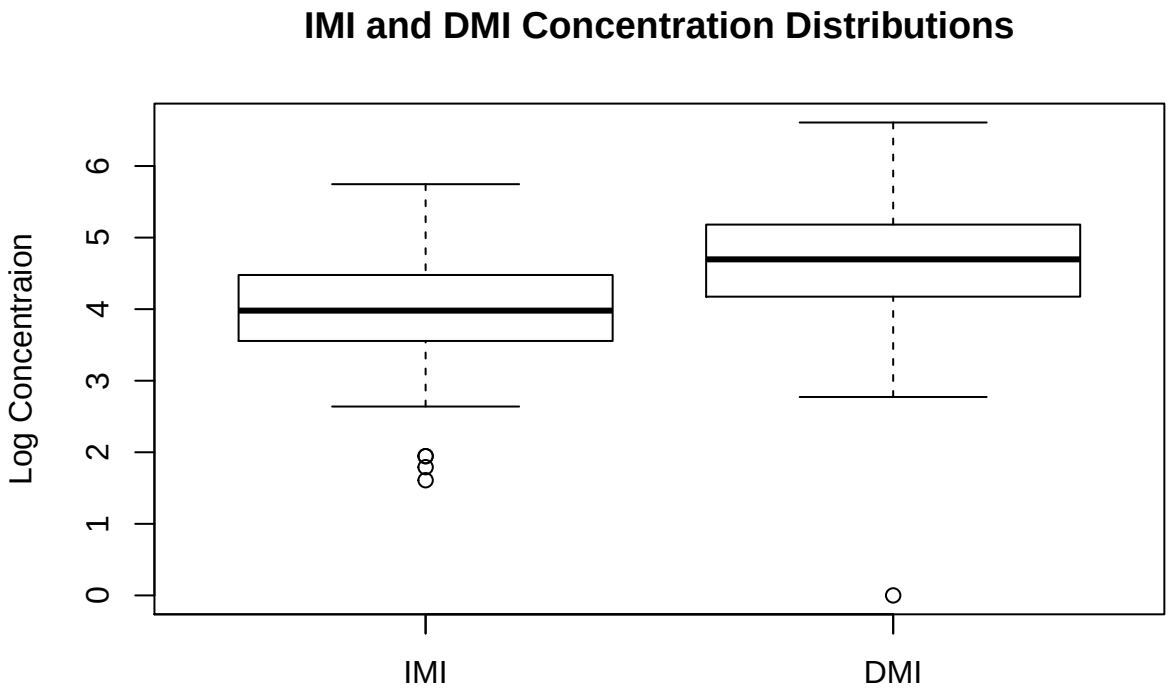
The female test subjects are overwhelmingly higher than the males (nearly double!).

Endogenous vs Reactive (Depression Type)



Most Depression cases in the test population are Endogenous (i.e. not a reaction to an environmental event).

DMI and IMI Concentrations



Generally tight distribution *especially around 25% to 75%) for both with little outlines with both distribution looking nearly identical sans the shift up with DMI. It seems that after being processed as DMI, the concentration of the antidepressant in the blood increases.

Models

General Diagnostic notes

For diagnostics, all models start with 1000 samples with no thinning and no burn in discard. The models are then diagnosed using the following techniques/measurements:

- Trace plots: Used to observe general mixing via parameter values, aim to have random caterpillar like shape.
- Effective size: Used to observe correlation effect on data by check how effective the samples are compared to independent data. Aim to have them consistently close and somewhere near half the samples.
- Gelman Rubin: Used to check convergence of model through comparing chains, aim to have value of 1 or very close to 1.
- auto-correlation plots: Useful to check need for thinning and thinning effect by checking sample correlations between each other, lags presented here should remain consistently low.

Linear Regression

In this method a relationship between a predictor x and a response y is established through a Linear Function with a coefficient the predictors and the intercept.

Data Cleaning and Preparation

Weeks

There are going to be two weeks variables, the initial week number for Auto-regression and Gaussian Process, and another or on whether it is a placebo week or not for linear regression.

Standardisation and Predictor Separation

The response (Hamilton index) and predictors (everything else) are separated for the modelling stage. Predictors (x) for linear regression use placebo indicator for weeks instead of actual weeks (not a time series). All non indicator predictors are standardised using a to ensure a fair impacts between variables and easier uninformative prior selection.

Basic Multiple

Multiple Linear Regression is a simple extension of Linear regression where multiple predictors each with their own coefficients are introduced.

Modelling and Diagnostics

The model was constructed in JAGS using a multiple linear regression based on a normal model using uninformative priors gamma(0.001, 0.001) for precision and $N(0, 10^6)$ for coefficients. The final model is truncated to values between 0 and 60 since this is the range for the Hamilton scores.

For diagnosis, The trace plot showed good catepillar like shape and Gelman Rubin scores were close to 1 (highest 1.04 for upper b0), but the sample sizes of b1 and b0 were poor (around 600 compared to 1400-3800 for other parameters) and their auto correlation plots shows signifcant lag till around 7-8. A burn in of a 1000 and a thin of 8 seemed to have fixed the problem (no signifcant initial lag and sample sizes around 3600-4500).

Posterior Summaries (Highest Interval Densities)

Table 2: Multiple Linear Model Posterior Summary

	lower	upper
b[1]	-2.091977	1.4062664
b[2]	-2.725037	0.5185505
b[3]	1.548595	5.1566424
b[4]	-1.086410	0.5350905
b[5]	-3.077219	-1.4071596
b0	21.340356	24.4720642
sd	6.068563	7.1460415

The highest interval of most coefficient seems rather wide, which could be due to factors like uncertainity about their effect in the model and/or due to limited data samples. Nonetheless, the results show that $b5$ shows the most negative effect on the score (more IMI in blood makes you less depressed) and $b3$ shows the most positive effect on the score (being in a placebo week makes you more depressed). The Standard Deviations are relatively wide around 6-7, and a $b0$ of around 21-25 suggests that if all the impacts of other parameters adds up to zero the expected score is that.

DIC

Table 3: DIC for Multiple Linear Model

Deviation	Penalty	Penalised Dev.
1654	7.11	1661

The DIC score on its own is hard to interpret, so it is better to first take a look at the DIC scores of the other models first.

Term Interactions

It is possible to extend Linear Models with term interactions. Term interactions are non-linear terms that extend linear model via relationships between variables (usually decided by a coefficient c). A potential interaction to add is between IMI and DMI since as stated in the data exploration stage these two predictors seem to interact, which is not surprising considering that DMI is the processed form of IMI.

Modelling and Diagnostics (IMI and DMI)

The model was constructed in JAGS similarily to the previous multiple linear model, but a new prior for the coefficient c controlling the effect of the interaction is introduced with a uninformative prior of $N(0, 10^6)$.

When it comes to the diagnostics, the trace plots here showed burn in issues with initial values changing rapidly suggesting that perhaps the effect of coefficient c is poorly understood initally and some burn in is needed. Other diagnostics showed similar results however to the previous model, as Gelam showed convergence, small b0 and b1 effective sizes were observed (around 500) and lastly lag of up to 9 was observed in the auto-correleation plots with $b0$ and $b1$ being the worst. Nonetheless, the addition of burn in of 2000 and idential thinning to the previous model (8), the issues with the trace plots were gone and the effective sizes were now similar to the previous model being around 3600-4150.

Posterior Summaries (Highest Interval Densities)

Table 4: Linear Model with Interaction Terms Posterior Summary

	lower	upper
b[1]	-1.996307	1.5655897
b[2]	-2.860129	0.3864148
b[3]	1.466682	5.2635190
b[4]	-1.174419	0.5929163
b[5]	-3.148445	-1.4138289
b0	20.985744	24.3205344
sd	6.057183	7.1929826

The posterior summaries seem similar in fact near identical, with wide coefficients, strong negative b5 and strong positive b3, a similar sd and b0.

Comparison with Multiple Linear Regression (DIC)

Table 5: DIC for Multiple Linear Model With IMI and DMI Interaction Terms

Deviation	Penalty	Penalised Dev.
1655	8.07	1663

Unsuprisngly, the similarities in the posterior translated to similar DIC results which showed penalised deviance that is within room for error. It seems the interaction did not cause any significant differences to the model! This is perhaps because the interaction effect between IMI and DMI is not supported by the available data.

Hierarchical

Another possible extension to linear models is via hierarchies. This is done by considering that the data consists of groups which have different parameters. In this example we can consider that every individual is a different group (each person is different), which according to the wide range of scores observed in the exploration stage and the uncertainity in the coefficients of previous models might be a reasonable assumption.

ID Preparation

To make accessing users to find their groups easier the IDs are converted to (1,2, 3 . . .) counting form using factors and levels.

Modelling and Diagnostics

Again, the model was constructed in JAGS similarily to the simple multiple linear model, but a new priors for the coefficients c controlling the effect of the groups is introduced with a uninformative prior of $N(0, gamma(0.001, 0.001))$.

For the diagnosis, the trace plots showed similar behaviour to the previous model, with most parameters changing rapidly initally suggesting a need for burn in and a misundertsood effect of parameter c. When it comes to the other diagnosis methods, they seem to show significant issues with b1, b2 and b0 with tiny effective size (around 40-90!), poor convergence (uppers of 1.19 for b1, 1.09 for b2 and 1.21 for b0) and lastly significant lag (around 20-26 for most b parameters including b1,b2 and b0).

Hence, to resolve these issues, a burn in of 3000, a thinning of 40 was and increase of samples sizes of 2000 introduced, which resolved most of the issues as the inital lag was one or two at worst and trace plot and effective sizes (now more balanced around 5500-8000) outputted betters results.

Posterior Summaries (Highest Interval Densities)

	lower	upper
b[1]	-3.572464	2.3931360
b[2]	-3.902135	1.5330715
b[3]	2.237850	4.7419565
b[4]	-1.101674	1.3318887
b[5]	-3.006797	-0.7754718
b0	20.192268	25.6376371

While again similar to previous models, these results some noticiable changes with some parameters, as $b0$ is now wider and the effect of $b5$ and $b3$ is now weakend suggesting that people react differently to the IMI drug and some are not as depressed during placebo week.

Comparison with Multiple Linear Regression (DIC)

Table 7: DIC for Hierarchical Multiple Linear Model

Deviation	Penalty	Penalised Dev.
1441	60.04	1501

This model seems to do noticibly better than the previous two with a lower penalised deviance of around 1500 compared to the previous two of around 1660. This proves that each person acts as a different group with different distrubution beahviour which explains the uncertainity.

AR(1)

AR models are time series model (allow dependence on responses of earlier times) that is based on a linear like model that predicts on a given number of previous responses with the help of "a" coefficients to control these previous responses effect. AR models also have another a0 coefficient usually referred to as a mean mu which control convergence point. In this example, AR models would suggest that scores from previous weeks effect other weeks, which would not be a difficult assumption to make since the depression severity of a previous week is likely to impact the next.

Since we are only dealing with 4 weeks, a AR model which uses the previous response only is a reasonable choice since there are not many previous responses to choose from!

Data Cleaning and Perparation (User Matricies)

However, a big problem with implementing AR models here arises because we need to use every user previous response not the previous response. And, as shown with the performance of the hierarchical model the individuals of the study are independent and using any other previous response would seriously negatively impact the model.

Hence, to ensure that we only use previous responses of the same users the Hamilton scores are stored in a matrix of rows of user IDs and columns of weeks which will be iterated through using a nested loop. To also ensure that other predictors will be available during the nested loop, the other predictors are also stored in similar matricies.

Modelling and Diagnostics

The AR(1) model is constructed in JAGS using an a1 coefficient based on a normal prior of $N(0, 0.35^2)$ to ensure it avoids values 1.0 and -1.0 which caused walking issues during diagnosis. The mu parameter is based on weakly informative a prior of $N(0, 10^6)$ that is truncated to (0,60) (recall Hamilton score range), the use of the informative prior is based on the hamiltion score ranges explored in the data exploration stage and was done because mu had a very wide range with an uinformative prior. The precision of the AR model is using an uninformative prior ofdgamma(0.001, 0.001). Finally, hierarchitcal behaviour is also introduced here since it proved to improve the model beahviour, the hierarchital c coefficient will be using a uninformative prior of $N(0, dgamma(0.001, 0.001)^-2)$ for every group as before.

During diagnosis, the model displayed similar behaviour to the previous model, with its parameters espcially a1 and hierachical standard deviation showing initial rapid changes shows a burn in discard requirement, but it seems most of the convergence and correleration issues moved to the hierartical standard deviation with very high upper interveral of 1.75, very poor effectve size of 50 and significant lag till around 90. Hence, a significant thinning of 100 was introduced with increased sample sizes of 3000 which made it converge and fixed most of the correlation issues (but still worse inital lag of around 5 and poor effective sizes of around 5000 vs 12000 for other parameters).

Posterior Summaries (Highest Interval Densities)

Table 8: AR(1) Posterior Summaries

	lower	upper
a1	0.7006733	0.9225959

	lower	upper
b[1]	-1.5135338	0.1610909
b[2]	-1.1840252	0.3800143
b[3]	-1.1994594	2.1529572
b[4]	-1.7337699	1.2965342
mu	1.2301889	20.6767105
sd	4.1144194	5.1878658
sd_hier	0.0174763	1.3305916

An a1 parameter of around 0.7-0.91 suggests a decyaing random walk converging to a given value. For the other parameters it seems that the mu parameter is quite wide, suggesting that it is difficult to ocme to a consensus about the average score. Interestingly, this model thinks the strongest negative impact on score can be the type of depression (being non-reactive), but that it can also be positively impacting as well (i.e. people who have reactive depression can be very depressed but also less depressed than non-reactive depression patients). Also, surpringly, the strongest positively impacting coefficient is now the sex, with female being more depressed (but sometimes this can be the other way around!).

When it comes to the impact of the drug itself, this model suggests it is mostly negative according to b1 and b2 (imi and dmi), but not as strong as other models suggested it is!

Comparison with Hierarchical Linear Regression (DIC)

Table 9: DIC for AR(1) Model

Deviation	Penalty	Penalised Dev.
922	8.78	931

This model seems to perform significantly better than all the other models introduced here! With DIC penalised scores around 930 compared to the other models (around 1500-1660). This not only shows that there is indeed evidence of dependence between results of different weeks, but also that the assumptions made on the last section based on the summaries might also be true!

Conclusion

Appendix

Abbrevations

IMI - antidepressant drug imipramine

DMI - desmethylimipramlne (Processed IMI)

AR - Auto Regressive Models

AR(1) - Auto Regressive Models (1st Degree)

JAGS - Just Another Gibbs Sampler MCMC based Bayesian sampler

MCMC - Markov chain Monte Carlo

DIC - Deviance Information Criterion

Functions Code

DIC Table

```
# Creates a table for DIC with a given caption
dic_table <- function(dic, caption) {

  mean_dev = round(c(sum(dic$deviance)))
  pen = round(c(sum(dic$penalty)), 2)
  mean_pen_dev = round(c(mean_dev+pen))

  dic.data = data.frame(mean_dev, pen, mean_pen_dev)

    kable(dic.data, col.names =
          c("Deviation", "Penalty", "Penalised Dev."),
          caption = caption)
}
```

Data and Exploration Code

Data Loading

```
load("Reisby.RData")
Reisby = as.data.frame(Reisby)
```

Correlation (Pairs Plot)

```
kable(cor(Reisby[,-1])) # no id
```

Hamilton Scores Histogram

```
hist(Reisby$hd, main = "Hamilton Index Scores Histogram",
     xlab = "Hamilton Score")
```

Sex Histogram

```
# Replace x axis here with male/female axis

hist(Reisby$female, main = "Sex Histogram",
     xlab = "Sex", xaxt = "n", ylim = c(0, 200))
axis(1, at=0:1, labels=c("Male","Female"))
```

Depression Type Histogram

```
hist(Reisby$reactive_depression, main = "Depression Type Histogram",
     xlab = "Depression Type", xaxt = "n", ylim = c(0, 150))
axis(1, at=0:1, labels=c("Endogenous","Reactive"))
```

DMI and IMI Boxplot

```
boxplot(Reisby$lnimi, Reisby$lndmi, main = "IMI and DMI Concentration Distributions",
        ylab = "Log Concentraion")
axis(1, at=0:2, labels=c("IMI","IMI", "DMI")) # uses 3 labels due to strange bug
```

Multiple Linear Model Code

Weeks Cleaining

```
placebo = c(ifelse(Reisby$week == 0, 1, 0))
```

Predictor Separation

```
x = Reisby[,-2:-3]
x$placebo = placebo
y = Reisby[,2]
```

Standarisation

```
scaled_dmi = (Reisby$lnimi - mean(Reisby$lnimi))/sd(Reisby$lnimi)
scaled_imi = (Reisby$lndmi - mean(Reisby$lndmi))/sd(Reisby$lndmi)

lin_x_scaled = x[, -2:-3]

lin_x_scaled$scaled_dmi = scaled_dmi
lin_x_scaled$scaled_imi = scaled_imi
```

Model

```
data = list(x = lin_x_scaled[,-1], y = y,
            n = nrow(lin_x_scaled),
            p = ncol(lin_x_scaled[,-1]))

# JAG model (as a string)

model_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)

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

}
"

# model construction and sampling
```

```
model_tc = textConnection(model_string)
model = jags.model(model_tc, data = data, n.chains = 4)

update(model, n.iter = 1000)
lin_samples = coda.samples(model,
                       variable.names = c("b0","sd","b"),
                       n.iter = 1000 * 8, thin = 8)
lin_dic = dic.samples(model, n.iter = 1000 * 8, thin = 8)
```

Summary

```
hpd = HPDinterval(lin_samples)
kable(hpd[[1]],
      caption = "Multiple Linear Model Posterior Summary")
```

DIC

```
dic_table(lin_dic, "DIC for Multiple Linear Model")
```

Linear Term Interactions Code

Model

```
data = list(x = lin_x_scaled[,-1], y = y,
            n = nrow(lin_x_scaled),
            p = ncol(lin_x_scaled[,-1]))

# JAG model (as a string)

model_string = "
model {
  b0 ~ dnorm(0, 1E-6)
  c ~ 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)

  for (i in 1:n) {
    y[i] ~ dnorm(mu[i], tau)
    mu[i] = b0 + inprod(b, x[i,]) + c * x[i, 4] * x[i, 5]
  }

}
"

# model construction and sampling

model_tc = textConnection(model_string)
model = jags.model(model_tc, data = data, n.chains = 4)

update(model, n.iter = 2000)
terms_samples = coda.samples(model,
                        variable.names = c("b0","sd","b"),
                        n.iter = 1000 * 9, thin = 9)
terms_dic = dic.samples(model, n.iter = 1000 * 9, thin = 9)
```

Summary

```
hpd = HPDinterval(terms_samples)
kable(hpd[[1]],
      caption =
      "Linear Model with Interaction Terms Posterior Summary")
```

DIC

```
dic_table(terms_dic,
          "DIC for Multiple Linear Model
          With IMI and DMI Interaction Terms")
```

Linear Hierarchical Code

ID Perparation

```
h_lin_x_scaled = lin_x_scaled
h_lin_x_scaled$id = as.factor(h_lin_x_scaled$id)
levels(h_lin_x_scaled$id) = 1:length(levels(h_lin_x_scaled$id))
```

Model

```
data = list(x = h_lin_x_scaled[,-1], y = y,
            n = nrow(h_lin_x_scaled),
            p = ncol(h_lin_x_scaled[,-1]),
            n_subjects = length(unique(h_lin_x_scaled[,1])),
            subject = h_lin_x_scaled$id)

# JAG model (as a string)

model_string = "
model {
  b0 ~ dnorm(0, 1E-6)

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

  for (k in 1:n_subjects) {
    c[k] ~ dnorm(0, tau_hier)
  }

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

  for (i in 1:n) {
    y[i] ~ dnorm(mu[i], tau)
    mu[i] = b0 + inprod(b, x[i,]) + c[subject[i]]
  }

}
"

# model construction and sampling

model_tc = textConnection(model_string)
model = jags.model(model_tc, data = data, n.chains = 4)

update(model, n.iter = 3000)
h_samples = coda.samples(model,
                         variable.names = c("b0","sd","b", "sd_hier"),
                         n.iter = 2000 * 40, thin = 40)
h_dic = dic.samples(model, n.iter = 2000 * 40, thin = 40) # 13
```

Summary

```
hpd = HPDinterval(h_samples,
                  caption =
                 "Hierarchical Linear Model Posterior Summaries")
kable(hpd[[1]][-7:-72,])
```

DIC

```
dic_table(h_dic, "DIC for Hierarchical Multiple Linear Model")
```

AR Code

User Matricies Construction

```
scaled_dmi = (Reisby$lnimi - mean(Reisby$lnimi))/sd(Reisby$lnimi)
scaled_imi = (Reisby$lndmi - mean(Reisby$lndmi))/sd(Reisby$lndmi)

SReisby = Reisby[, -4:-5]
SReisby$scaled_dmi = scaled_dmi
SReisby$scaled_imi = scaled_imi

weekCount = count(SReisby, "id")
full_ids = subset(weekCount, freq == max(SReisby$week) + 1)
sub_SReisby = subset(SReisby, id %in% full_ids$id)
ids = unique(sub_SReisby$id)


score_mat = matrix(nrow = length(ids),
                   ncol = max(sub_SReisby$week) + 1)
imi_mat = matrix(nrow = length(ids),
                   ncol = max(sub_SReisby$week) + 1)
dmi_mat = matrix(nrow = length(ids),
                   ncol = max(sub_SReisby$week) + 1)
female_mat = matrix(nrow = length(ids),
                   ncol = max(sub_SReisby$week) + 1)
dep_mat = matrix(nrow = length(ids),
                   ncol = max(sub_SReisby$week) + 1)

for(i in 1:length(ids)){
  for(j in 0:max(sub_SReisby$week)){
    score_mat[i, j+1] = subset(sub_SReisby, id == ids[i] & week == j)$hd
    imi_mat[i, j+1] = subset(sub_SReisby, id == ids[i] & week == j)$scaled_imi
    dmi_mat[i, j+1] = subset(sub_SReisby, id == ids[i] & week == j)$scaled_dmi
    female_mat[i, j+1] = subset(sub_SReisby, id == ids[i] & week == j)$female
    dep_mat[i, j+1] = subset(sub_SReisby, id == ids[i] & week == j)$reactive_depression

  }
}
```

Model

```
### Modelling (AR(1))

# Reuse Hierarchical model ids
data = list(y = score_mat, x1 = imi_mat, x2 = dmi_mat, x3 = female_mat,
            x4 = dep_mat, nrow = nrow(score_mat), ncol = ncol(score_mat),
            p = 4)

model_string = "
model {

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

```
  }

  for (l in 1:nrow) {
    c[l] ~ dnorm(0, tau_hier)
  }

  tau_hier ~ dgamma(0.001, 0.001)
  sd_hier = pow(tau_hier, -0.5)

  for (i in 1:nrow) {
    for(k in 2:ncol){
      y[i,k] ~ dnorm (mu + a1 * (y[i, (k-1)] - mu) +
                 x1[i, k] * b[1] + x2[i, k] * b[2] +
                 x3[i, k] * b[3] + x4[i, k] * b[4] + c[i], tau)
    }
  }

  mu ~ dnorm(0, 1E-6) T(0,60)
  a1 ~ dnorm (0, 0.35^-2)
  tau ~ dgamma (0.001, 0.001)
  sd = pow (tau , -0.5)
}
"

model_tc = textConnection(model_string)
model = jags.model(model_tc, data = data, n.chains = 4)

## Sampling

update(model, n.iter = 3000)
ar1_samples = coda.samples(model, variable.names = c("mu", "a1", "sd",
                                                     "b","sd_hier"),
                                n.iter = 3000 * 100, thin = 100)
ar1_dic = dic.samples(model, n.iter = 3000 * 100, thin = 100)
```

Summary

```
hpd = HPDinterval(ar1_samples)
kable(hpd[[1]], caption = "AR(1) Posterior Summaries")
```

DIC

```
dic_table(ar1_dic, "DIC for AR(1) Model")
```