Bayesian Inference on PM10 particulate matter air pollution datasets for Heathrow and Haringey, London, between 2000 and 2004

Nicholas P.J. Harper, 700038738

Introduction

Air pollution data recording levels of PM10 particulate matter have been collected on a daily basis at sites at both Heathrow and Haringey, in London, throughout the period from 2000 to 2004, inclusive. These data can be found in the London_Pollution.csv file. The locations of the monitoring site are shown in the OpenMap image together with a London districts shapefile overlay (Figure 1).

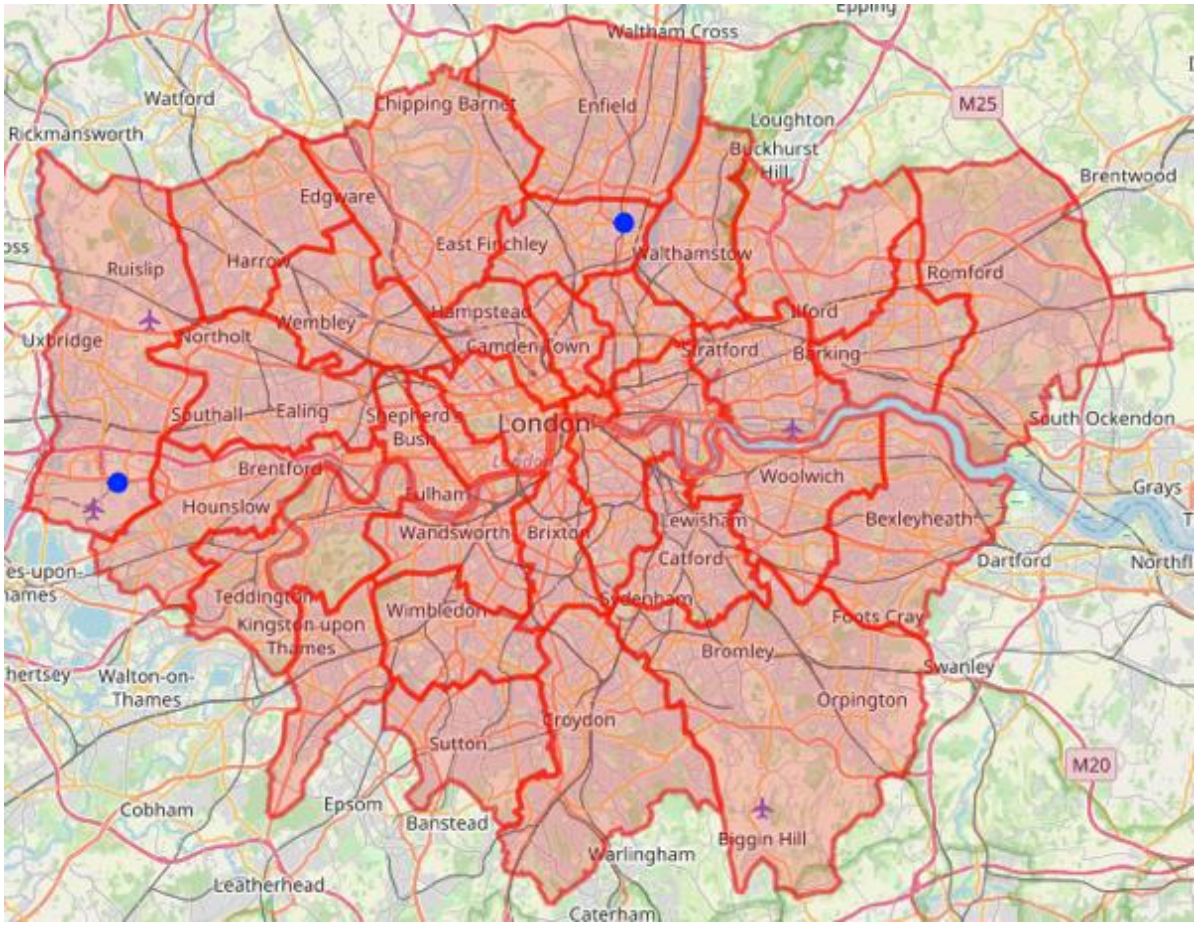


Figure 1: OpenMap image together with a London districts shapefile overlay showing the locations of the Heathrow and Haringey PM10 air pollution monitors in west London and north London, respectively

Summaries of the data from both Heathrow and Haringey are shown in Tables 1a and 1b, with mean values in the low 20s and inter-quartile ranges between 15 and up to 25, although the data tend to be a little higher at Heathrow compared to Haringey.

An analysis of the times of when the missing data (NA) occurred at each site has been undertaken, with facet-wrap bar charts produced to show the number of days of missing data for each month of each year at each site (figures 2 and 3). These intervals of missing data may reflect times when the monitor was either not working, was being maintained or was being upgraded. The details of the patterns in these data are described here.

At Heathrow, the frequency of NA results by month in Heathrow (see Figure 2) shows that there were very few missing results during January-October 2000, with 6 in May, 5 in October and only 2 or less in any other month. However, there were no results recorded during November-December 2000 and this lack of results continued throughout January-April 2001. During the rest of 2001, there are 8 missing results in April, 12 in October and only five or less in any other month. During 2002, there were only four or less in any month. During 2003, all results are missing for January-May and there are 12 missing results during June. Following this, there are no more than 3 missing results for any month thereafter during 2003 and 2004.

summary.heathrow	
Min.	3.10000
1st Qu.	15.50000
Median	19.50000
Mean	21.72375
3rd Qu.	25.90000
Max.	69.90000
NA's	404.00000

summary.haringey	
Min.	5.90000
1st Qu.	14.80000
Median	18.50000
Mean	20.23564
3rd Qu.	23.60000
Max.	65.90000
NA's	278.00000

Table 1: Statistical summaries of the PM10 particulate matter air pollution during the period 2000-2004 for: (a) Heathrow (left); and (b) Haringey (right). Note the number of NA (missing data) values at each site.

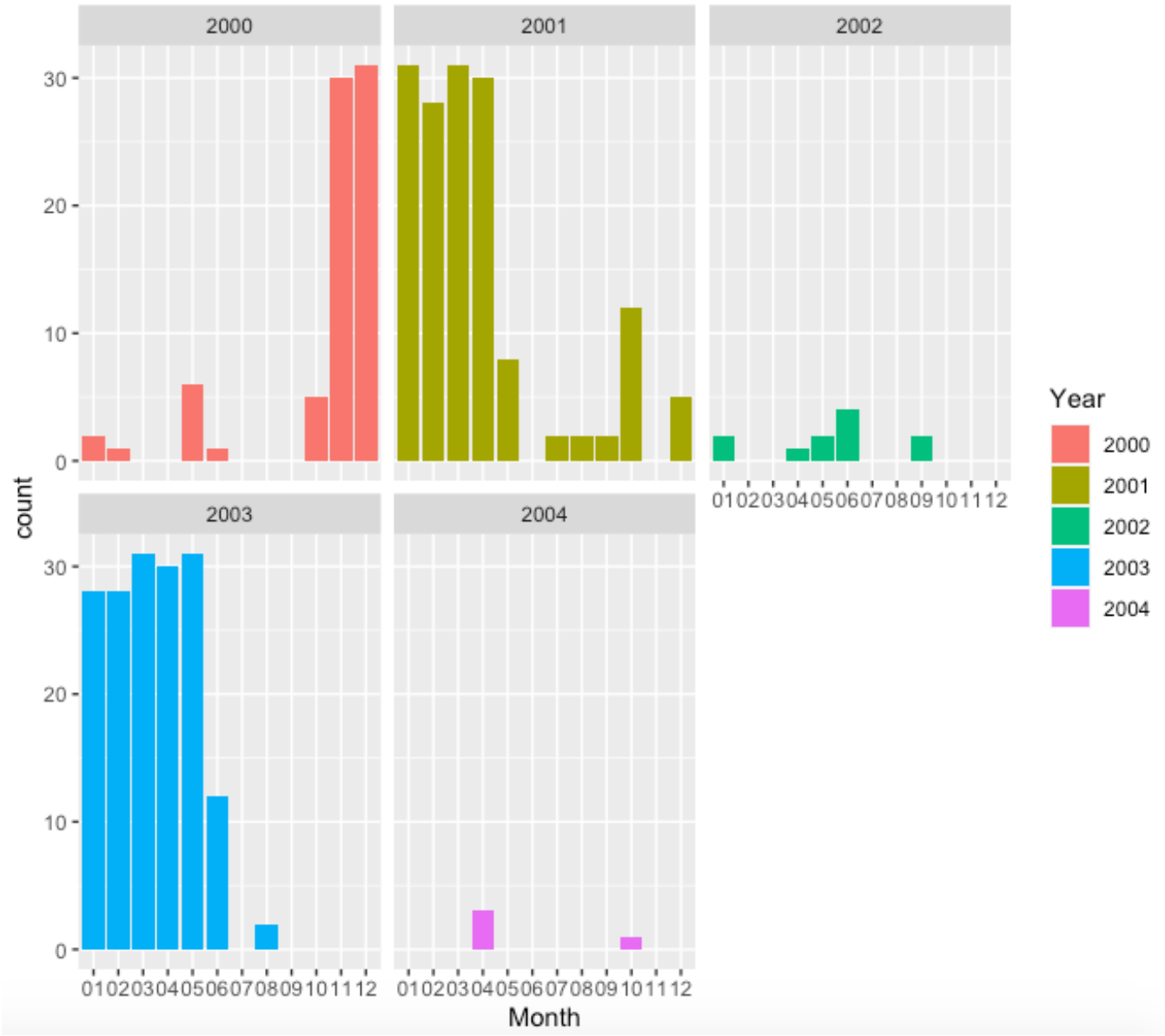


Figure 2: Facet wrap bar charts showing the number of days of missing PM10 air pollution data for each month of each year between 2000 and 2004, inclusive, for the monitor at Heathrow

The frequency of NA results by month in Haringey (see Figure 3) shows that there were very few missing results during 2000, with 10 in March and 9 in May, and only two or less in any other month. During 2001, there were only 2 missing results in total, which was during February. During 2002, there were only 5 missing results during January-August, however all the results are missing from September to December. During 2003, the missing results continue throughout January-March and most of April. Following this, there are no more than 3 missing results for any month thereafter during 2003 and 2004.

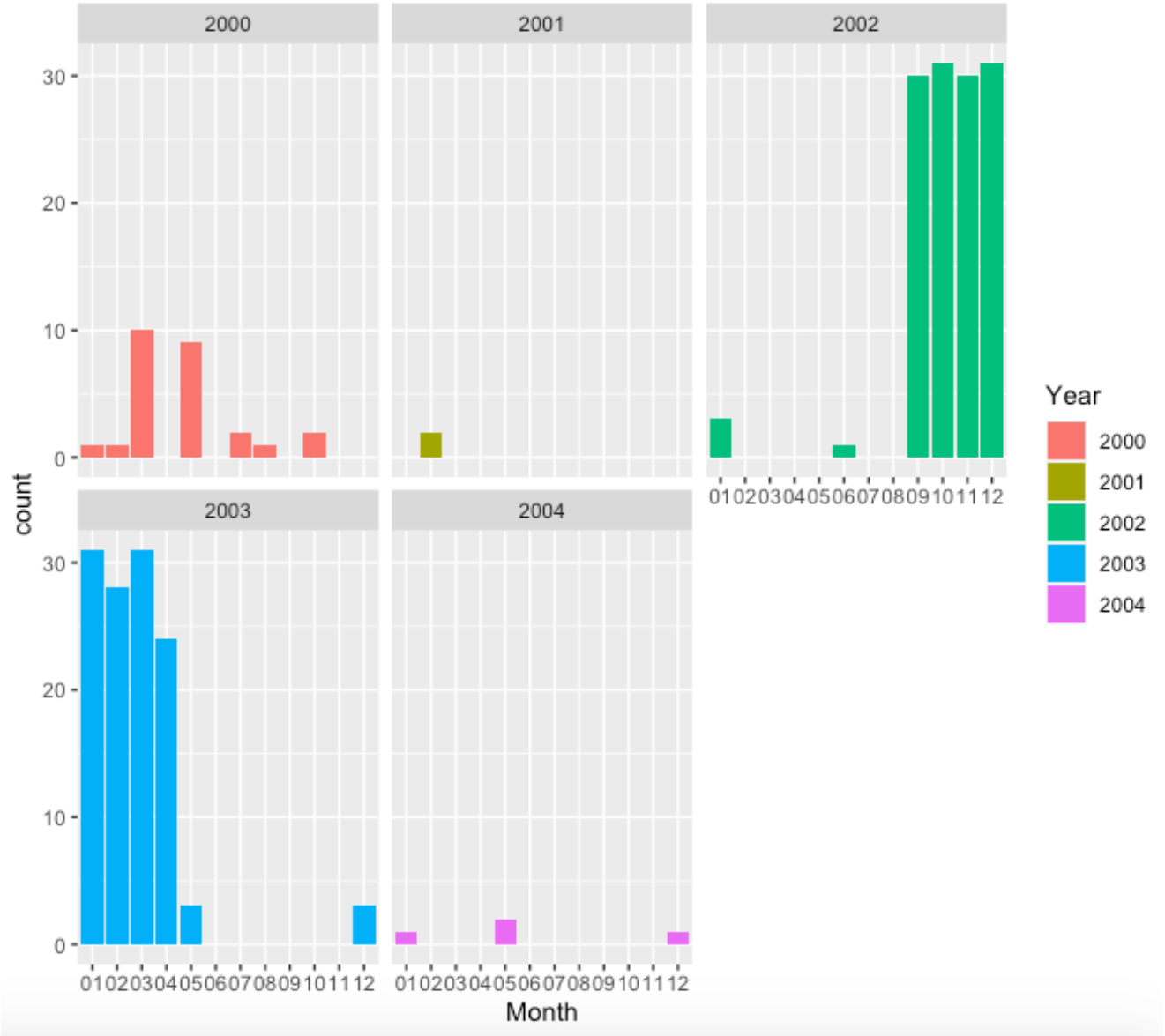


Figure 3: Facet wrap bar charts showing the number of days of missing PM10 air pollution data for each month of each year between 2000 and 2004, inclusive, for the monitor at Haringey

Further details of the missing data can be interpreted by plotting the PM10 data for each site against time, where the missing data are coerced to zero (figures 4 and 5). As with the facetted bar charts, this shows that there are two major periods of missing data at Heathrow and only one period at Haringey.

To understand these data for both sites better, it needed to be appreciated that measurements recorded may not fully represent the “true” values of PM10 air pollution due to range of factors, such as monitor calibration, resolution of data recorded, time of day that the data were recorded. Therefore, there may be some variance or difference between the measured value and what the actual “true” value may be. Also, as there are missing data where the actual “true” air pollution value is not known. Consequently, Bayesian inference will be performed on the data using different random walk models. The details of these methods are provided below:

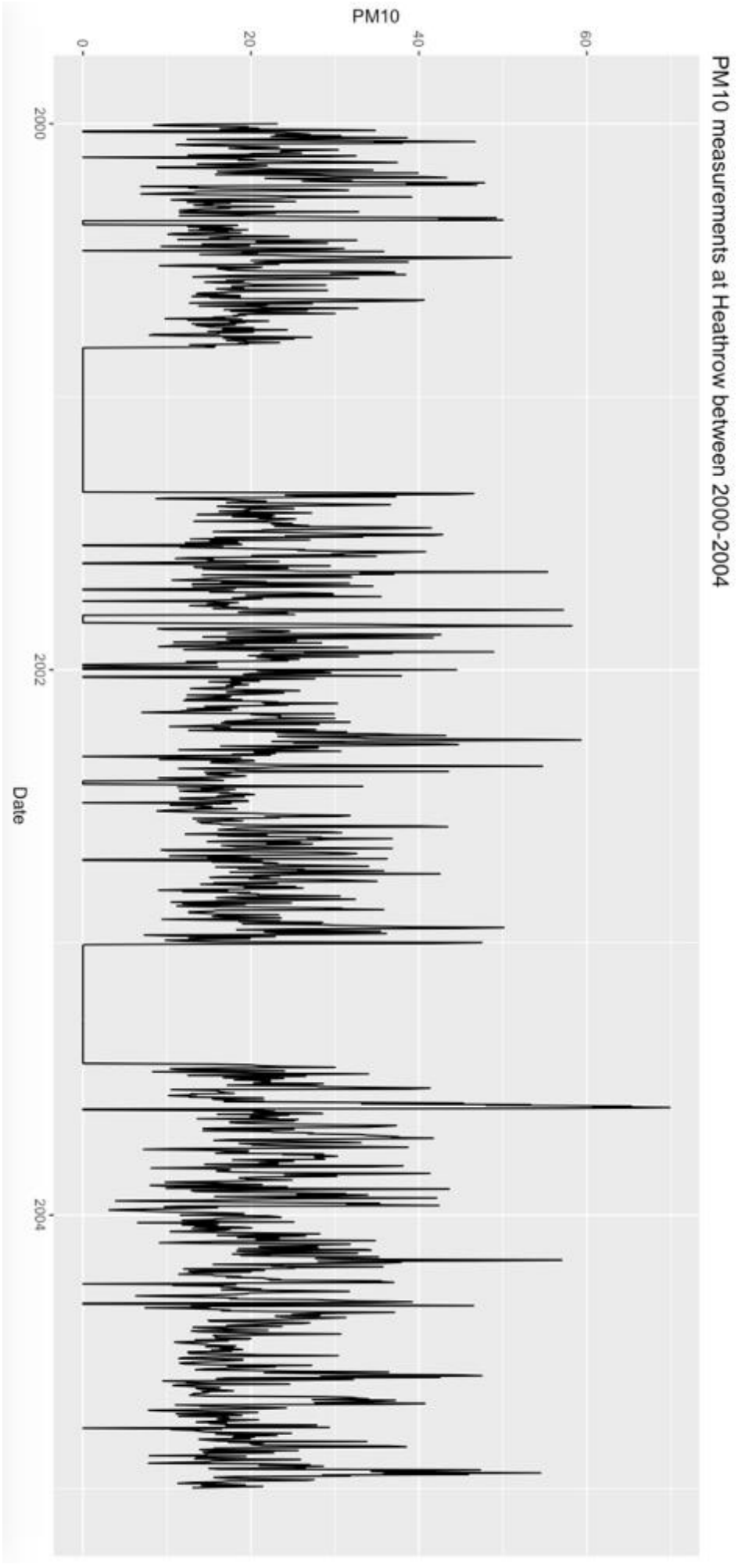


Figure 4: PM10 data recorded at Heathrow between 2000 and 2004, with periods of missing data shown as zero

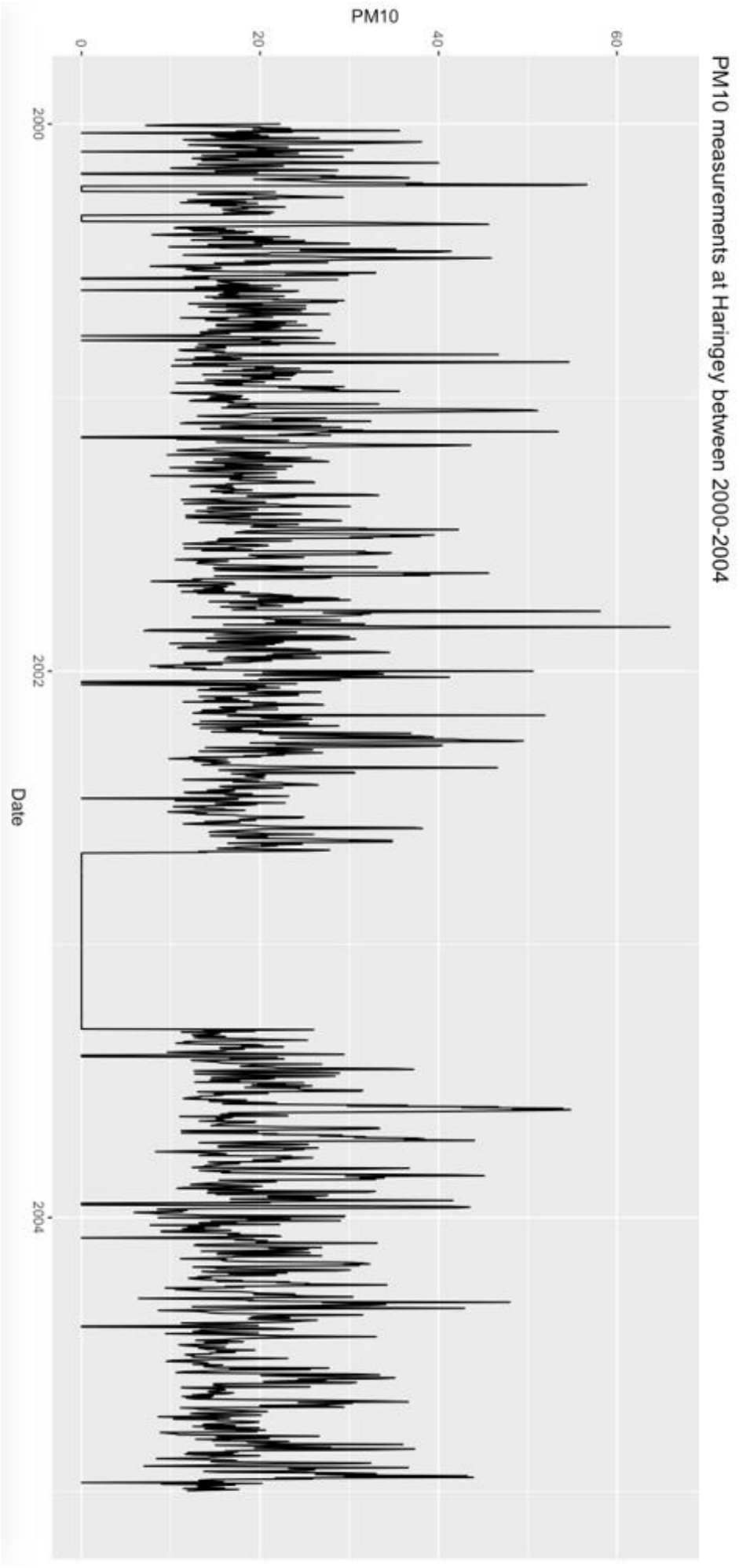


Figure 5: PM10 data recorded at Haringey between 2000 and 2004, with periods of missing data shown as zero

Methods for Random Walk model analysis and Bayesian inference

Random walk processes of order 1 (RW1) and order 2 (RW2) have been tested on the Heathrow and Haringey datasets to assess how applicable they are to model the data. Bayesian inference is performed in the JAGS package and each model approach is described as to how it was set up to be used in JAGS across the different model runs.

In the random walk 1 model: (1) the recorded measurement for one day ($Heathrow_t$ or $Haringey_t$) is related to the "true" measurement for that day (Y_t or Z_t) by the normal distribution with a vague precision (tau_i); and (2) the "true" measurement for one day (Y_t or Z_t) is related to the "true" measurement for the previous day (Y_{t-1} or Z_{t-1}) by the normal distribution with a vague precision (tau_i). The latter condition here means that the "true" measurement on the first day (Y_1 or Z_1) is not known and therefore needs to be initiated, in this case by a normal distribution with zero mean and precision of 0.001. In order to differentiate and to avoid confusion, the use of the symbol Y_t is to link with the Heathrow data ($Heathrow_t$), whilst the symbol Z_t is to link with the Haringey data ($Haringey_t$). Also, the vague prior distributions include descriptions of the precision (tau_i) for both models each given by a gamma distribution with equal parameters both of 0.001 as well as a link between the variance ($sigma^2_i$) as the inverse of the precision (tau_i). This is given in the general model formula here, where (i = 1, 2, …, 28):

$$
\begin{gathered}
jags.mod. < location > = function(\{ \\
Y[1] \sim dnorm(0, 0.001) \\
for(t\ in\ 2{:}N1)\{ \\
< location > [t] \sim dnorm(Y[t], taui) \\
Y[t] \sim dnorm(Y[t-1], taui) \\
\} \\
\text{\# vague priors} \\
taui \sim dgamma(0.001, 0.001) \\
sigma2_i = 1/taui \\
taui \sim dgamma(0.001, 0.001) \\
sigma2_i = 1/taui \\
\}
\end{gathered}
$$

In the random walk 2 model: (1) the recorded measurement for one day ($Heathrow_t$ or $Haringey_t$) is related to the "true" measurement for that day (Y_t or Z_t) by the normal distribution with a vague precision (tau_i); and (2) the "true" measurement for one day (Y_t or Z_t) is related to a function of the "true" measurement for the previous day ($2*Y_{t-1}$ or $2*Z_{t-1}$) minus the "true" measurement for 2 days previously (Y_{t-2} or Z_{t-2}) by the normal distribution with a vague precision (tau_i). The latter condition here means that the "true" measurement on the first (Y_1 or Z_1) and second (Y_2 or Z_2) days are not known and therefore need to be initiated, in both cases by a normal distribution with zero mean and precision of 0.001. Again, the vague prior distributions include descriptions of the precision (tau_i) for both models each given by a gamma distribution with equal parameters both of 0.001 as well as a link between the variance ($sigma^2_i$) as the inverse of the precision (tau_i). This is given in the general model formula here, where (i = 1, 2, …, 28):

$$
\begin{gathered}
jags.mod. < location > = function(\{ \\
Y[1] \sim dnorm(0, 0.001) \\
Y[2] \sim dnorm(0, 0.001) \\
for(t\ in\ 3{:}N2)\{ \\
< location > [t] \sim dnorm(Y[t], taui) \\
Y[t] \sim dnorm(2 * Y[t-1] - Y[t-2], taui) \\
\} \\
\text{\# vague priors} \\
taui \sim dgamma(0.001, 0.001) \\
sigma2_i = 1/taui \\
taui \sim dgamma(0.001, 0.001) \\
sigma2_i = 1/taui \\
\}
\end{gathered}
$$

In order to test the forecasting ability of the two random walk models on the Heathrow data ($Heathrow_t$) for the first week of 2004, the root mean square error (RMSE) value ($rmse.Y_t$) is modelled as another parameter to be estimated, with an extra line of code added, into a model run with each model. The RMSE compares the "true" measurement against the estimated measurement for the same day and so the estimated measurement can be interpreted as the

recorded measurement ($Heathrow_t$), with the mean recorded value (mean.heathrow) given on days of missing data (RW1: jags.mod.heathrow7; RW2: jags.mod.heathrow8), whose results are described in this report. Alternatively, RMSE estimated measurement could be interpreted as the mean "true" measurement value (mean.Y) (RW1: jags.mod.heathrow5; RW2: jags.mod.heathrow6) but these results are not described here. With the models that include the RMSE calculations, to differentiate and to avoid confusion with the other "true" measurement parameters (Y_t), the symbol ($Y2_t$) is used instead. This extra line for comparing to the mean is given by:

$$rmse.Y[t] = sqrt(sum((mean.Y - Y2[t])^2 / N2))$$

Alternatively, this extra line for comparing to the recorded measurement is given by:

$$rmse.Y[t] = sqrt(sum((Heathrow2[t] - Y2[t])^2 / N2))$$

Also, the third and fourth random walk Haringey models use informative priors on the Haringey data ($Haringey_t$) from the first and second Haringey models, respectively. The fifth and sixth random walk Haringey models use informative priors on the Haringey data ($Haringey_t$) from the first and second Heathrow models, respectively. The values taken are the mean and variance of each posterior $sigma^2_i$ estimate from these models, transformed into precision estimates and applied to the precision estimates (tau_i) of the new Haringey models using a normal distribution. With the models that include the informative priors, to differentiate and to avoid confusion with the other "true" measurement parameters (Z_t), the symbol ($Z2_t$) is used instead. This means that the prior distributions for the last two Haringey models are given by:

informative priors
$$taui \sim dnorm(0.064, 0.020)$$
$$sigma2_i = 1/taui$$
$$taui \sim dnorm(0.040, 0.064)$$
$$sigma2_i = 1/taui$$

To run the JAGS model (of the form jags.mod.<location>), the recorded measurements ($Heathrow_t$ or $Haringey_t$), number of datapoints (N) and the initial values of both the first "true" measurement (Y_1 or Z_1) and of the missing data and put in, whilst the parameters to return are the variances ($sigma^2_i$) of the two models and the "true" measurements, with a parameter for each day (i.e. N-value of that model). Each JAGS model also runs 2 chains for 10000 iterations and "burns" the first 5000 iterations (burnin) of each chain to assist with model convergence for each parameter. All the information on the model inputs, initial values, parameters to save as well as chain, iteration and burnin numbers is put into a new fitted model that JAGS runs (of the form jags.mod.fit.<location>).

When each fitted model has run, an assessment of a selection of the model parameters, in particular the "true" measurement parameters (Y_t or Z_t) to assess what their mean, standard deviation, potential scale reduction factor (Rhat) and effective sample size (n.eff) is. A value of Rhat ~ 1, and definitely < 1.1, suggests that the parameter will converge. From this, density plots of the $sigma^2_i$ parameters for each model are created and the JAGS model is converted into a Markov Chain Monte Carlo (MCMC) object. This allows for generation of a selection of traceplots showing which "true" measurement parameters (Y_t or Z_t) to assess the pattern of which parameters have converged as well as calculation of the gelman diagnostic (gelman.diag) to compare the within-chain and between-chain variances for each parameter. A table of the gelman diagnostic outputs is made to match the selection of traceplots.

Lastly, a ggplot is generated for the length of time determined for the analysis showing the recorded measurements ($Heathrow_t$ or $Haringey_t$) as points as well as the "true" measurement parameters (Y_t or Z_t) with the associated 95% credible interval boundaries. This provides an excellent visual of how well the random walk model relates to the recorded measurements ($Heathrow_t$ or $Haringey_t$) and how the model has coped with the missing data intervals.

Results for Random Walk analyses and Bayesian Inference for the Heathrow dataset

Random walk analyses and Bayesian inference is undertaken on the Heathrow dataset. Initially, the data have been wrangled to remove the 2004 data, so that analysis is only undertaken on data between 2000 and 2003, inclusive, which constitutes 1461 (N1) datapoints (2000 was a leap year). For the forecasting models, there are an additional 7 datapoints to represent the first week of 2004, taking N2 to 1468. The mean of the Heathrow data, without the missing data, is also calculated, so that it can be used to initiate the missing data.

First Heathrow model (jags.mod.heathrow)

The first Heathrow model (jags.mod.heathrow) is a random walk of order 1 (RW1) and included as output the parameters: $sigma^2_1$, $sigma^2_2$, and all Y_t. The Y_t parameters that have associated recorded measurements ($Heathrow_t$) show Rhat values of ~ 1, suggesting that the chains would converge, whilst the Y_t parameters that have associated missing data show Rhat values > 1.1, suggesting that they would not converge. The density plots show that there are some minor differences in the estimations of $sigma^2_1$ and $sigma^2_2$ (Figures 6a and 6b) for each chain, which may relate to how each chain handled the missing data.

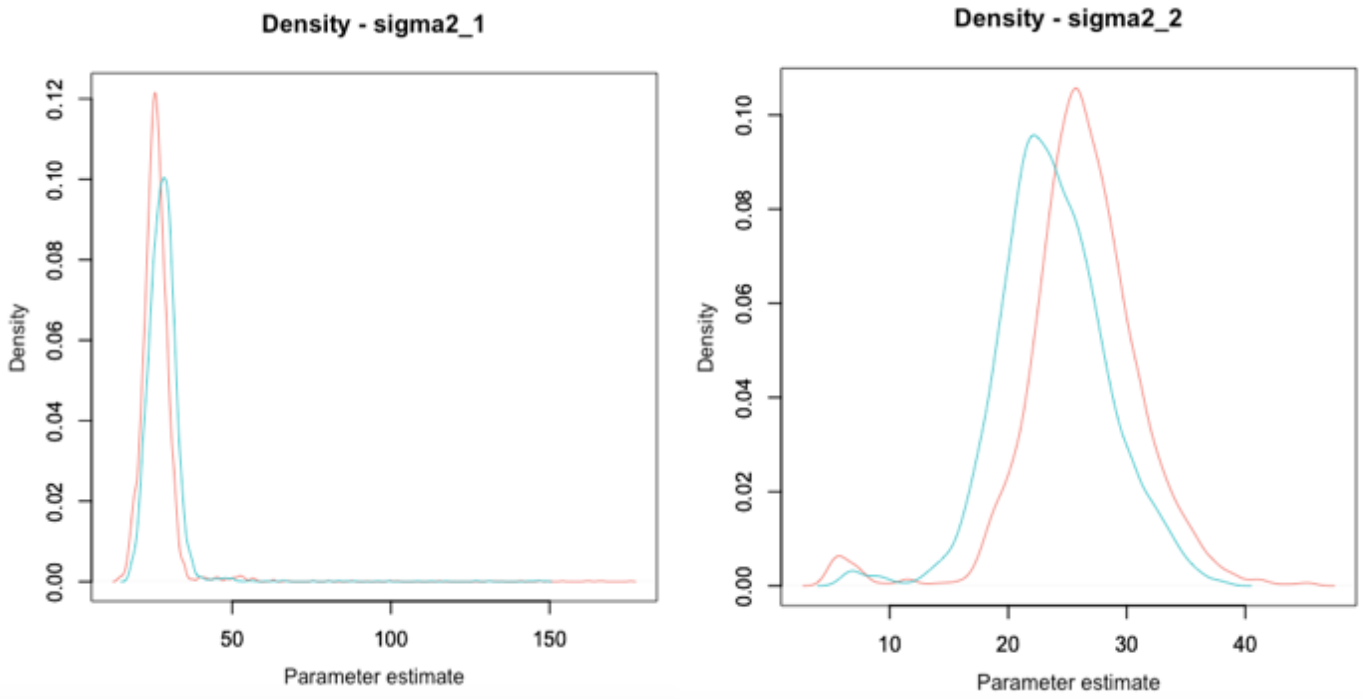
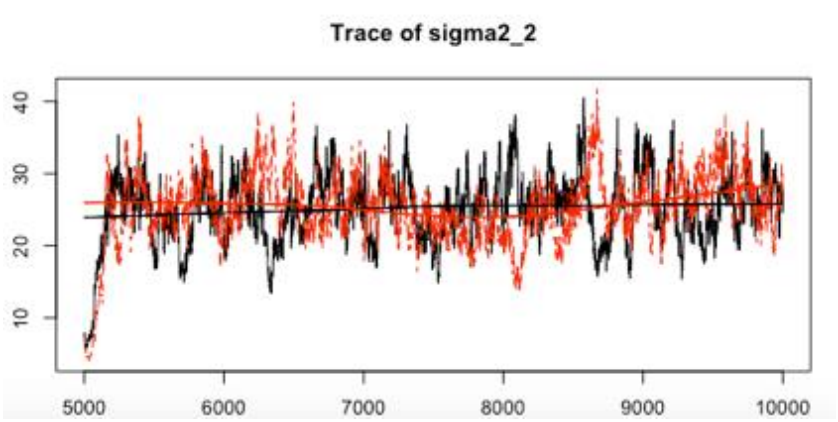
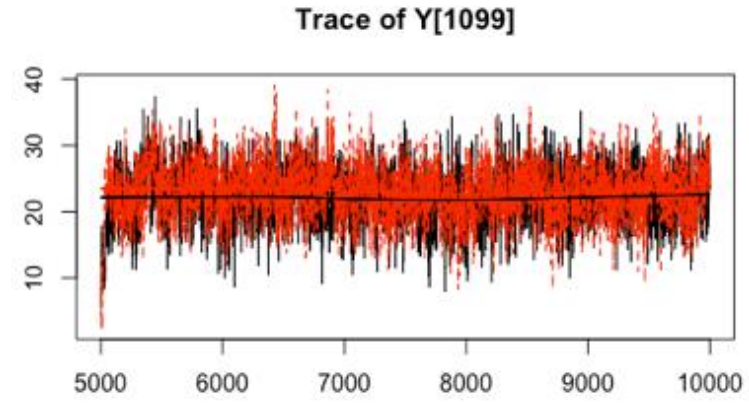


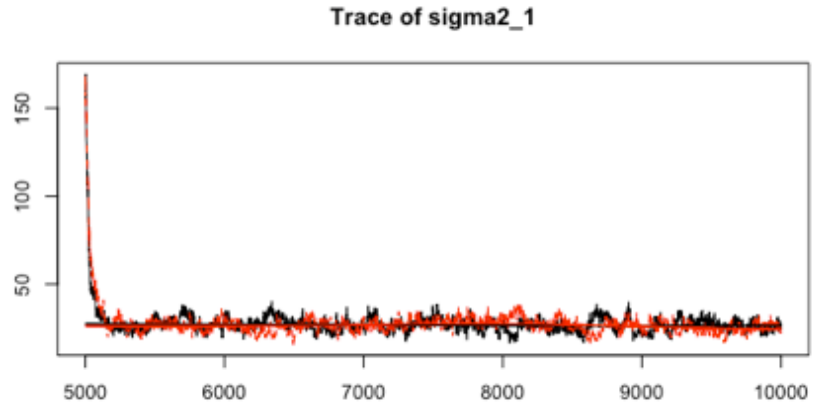
Figure 6: Density plots showing that the 2 chains have not quite converged for RW1 model: a) $sigma^2_1$, and b) $sigma^2_2$. This suggests that parts of the data cause difference variances in the outputs, which may result from the intervals of missing data.

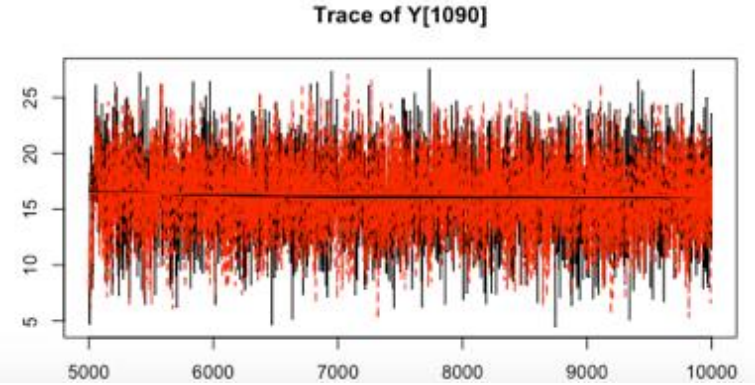
Consequently, 12 traceplots have been chosen. The first 3 show that amount of converge between the chains for deviance, $sigma^2_1$ and $sigma^2_2$ area all poor in an RW1 model. The next 9 show how the RW1 model chains converge for "true" measurement (Y_t) parameters that have associated recorded measurements ($Heathrow_t$), namely Y1 and Y1090, lose convergence as the "true" measurements encounter the missing data, namely in Y1099 and Y1100, stop converging in the missing data, namely Y1110 and Y1250, regain convergence as the "true" measurements encounter the recorded measurements again, namely Y1259 and Y1260, and fully converge once again with the recorded measurements, namely Y1265 (Figure 7).

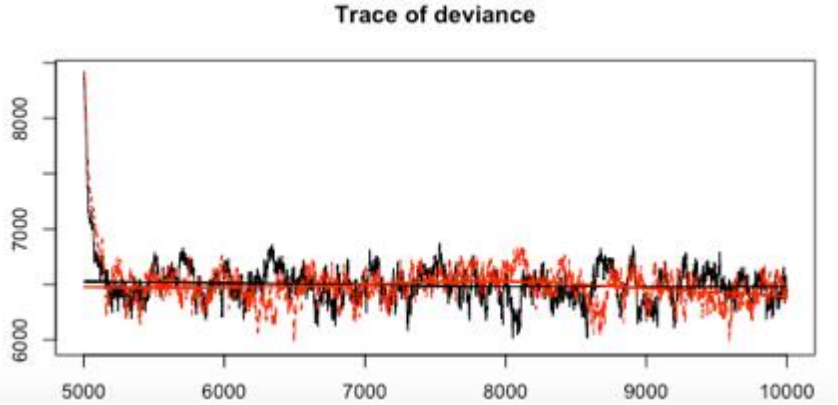
The table of the gelman diagnostics for selected parameters, namely of deviance, $sigma^2_1$, $sigma^2_2$, Y1 and Y1090 to Y1110 (Table 2) show that the point estimate and 95% CI estimate are > 1 whether they are associated either with recorded measurements ($Heathrow_t$) or with missing data. Further, the ggplot of the recorded measurements ($Heathrow_t$) as well as the "true" measurement parameters (Y_t) with the associated 95% credible interval boundaries shows that the random walk model approximates well to the recorded measurements ($Heathrow_t$) with tight 95% credible interval boundaries for 2000 to 2003 inclusive. However, where there are missing data intervals, the random walk model has no constraint and follows a random path until it encounters recorded measurements again, whilst the 95% credible intervals become very wide (Figure 8).

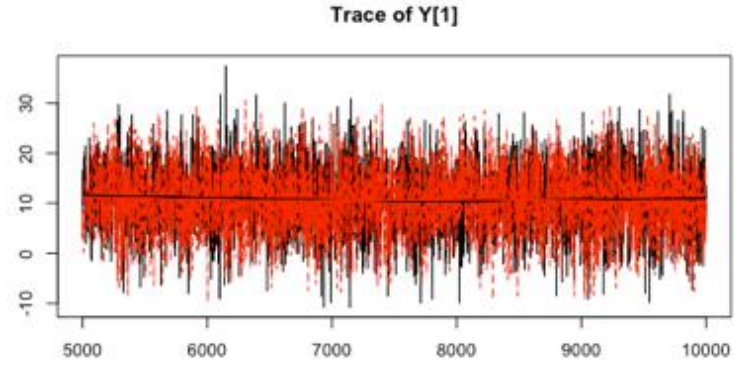

Trace of sigma2_2
40
30
20
10
5000
6000
7000
8000
9000
10000


Trace of Y[1099]
40
30
20
10
5000
6000
7000
8000
9000
10000


Trace of sigma2_1
150
100
50
5000
6000
7000
8000
9000
10000


Trace of Y[1090]
25
20
15
10
5
5000
6000
7000
8000
9000
10000


Trace of deviance
8000
7000
6000
5000
6000
7000
8000
9000
10000


Trace of Y[1]
30
20
10
0
-10
5000
6000
7000
8000
9000
10000

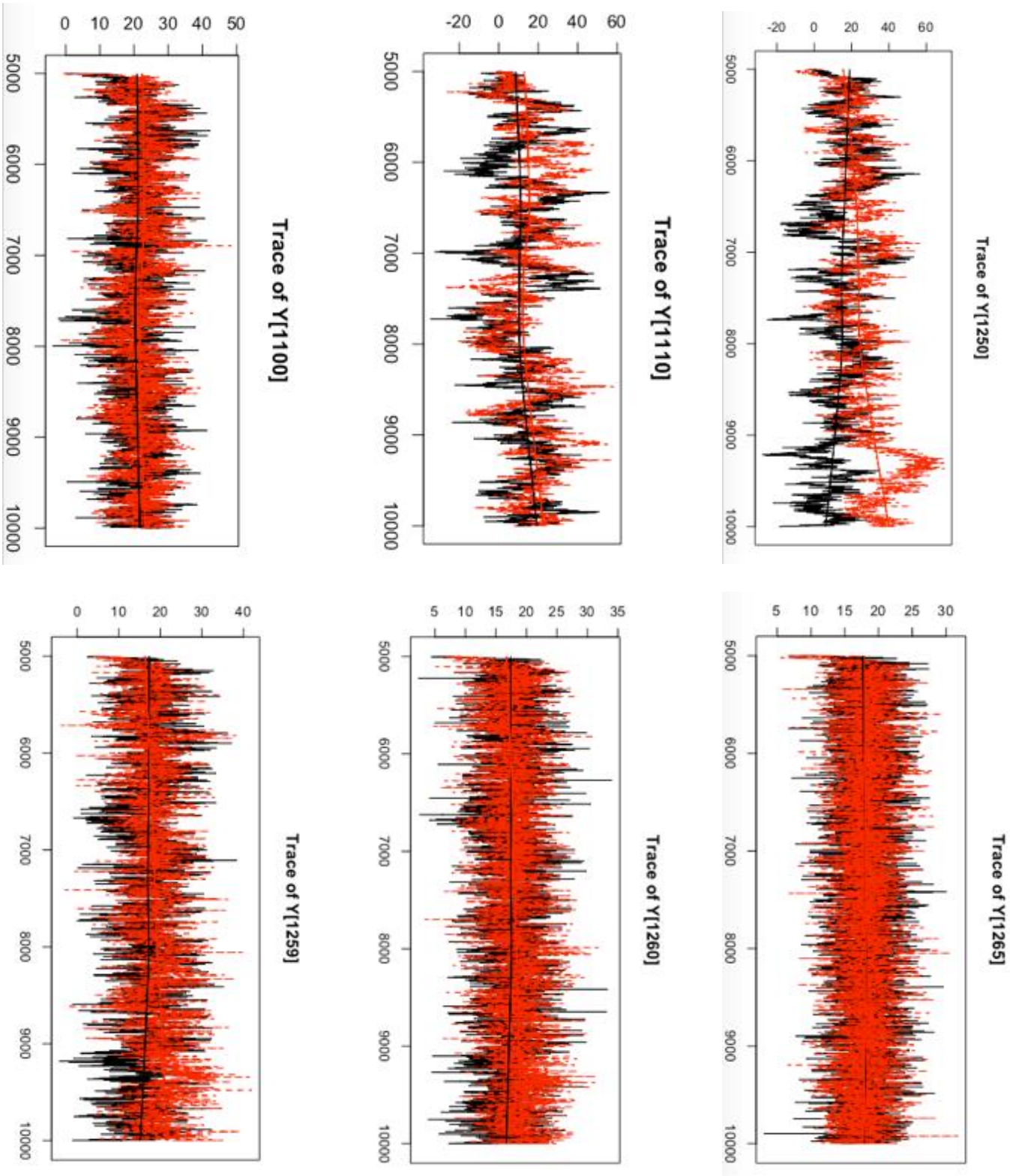


Figure 7: Progression of 12 traceplots showing how the 2 chains do not converge well for deviance, $sigma^2_1$ and $sigma^2_2$ in an RW1 model. The next 9 show how the RW1 model chains converge for "true" measurement (Y_t) parameters that have associated recorded measurements ($Heathrow_t$), namely Y1 and Y1090, lose convergence as the "true" measurements encounter the missing data, namely in Y1099 and Y1100, stop converging in the missing data, namely Y1110 and Y1250, regain convergence as the "true" measurements encounter the recorded measurements again, namely Y1259 and Y1260, and fully converge once again with the recorded measurements, namely Y1265

	Point est.	Upper C.I.
deviance	1.0027915	1.0038873
sigma2_1	1.0042921	1.0043886
sigma2_2	1.0018728	1.0050301
Y[1]	1.0002035	1.0013594
Y[1090]	1.0011389	1.0054175
Y[1091]	1.0003750	1.0021480
Y[1092]	1.0008113	1.0020929
Y[1093]	0.9999628	1.0002054
Y[1094]	0.9999832	1.0000131
Y[1095]	1.0006170	1.0012850
Y[1096]	0.9999606	0.9999606
Y[1097]	1.0004740	1.0006914
Y[1098]	1.0002631	1.0006411
Y[1099]	1.0015391	1.0076623
Y[1100]	1.0091385	1.0407337
Y[1101]	1.0223862	1.0948377
Y[1102]	1.0432818	1.1719636
Y[1103]	1.0719537	1.2739731
Y[1104]	1.1022984	1.3757576
Y[1105]	1.1300527	1.4666992
Y[1106]	1.1598985	1.5643914
Y[1107]	1.1825379	1.6366388
Y[1108]	1.2108592	1.7252333
Y[1109]	1.2405946	1.8186251

Table 2: Results table of the gelman diagnostics for selected parameters for RW1 model, namely of deviance, $sigma^2_1$, $sigma^2_2$, Y1 and Y1090 to Y1110 show that the point estimate and 95% CI estimate are ~1 when there are associated recorded measurements ($Heathrow_t$) and become > 1 when associated with missing data

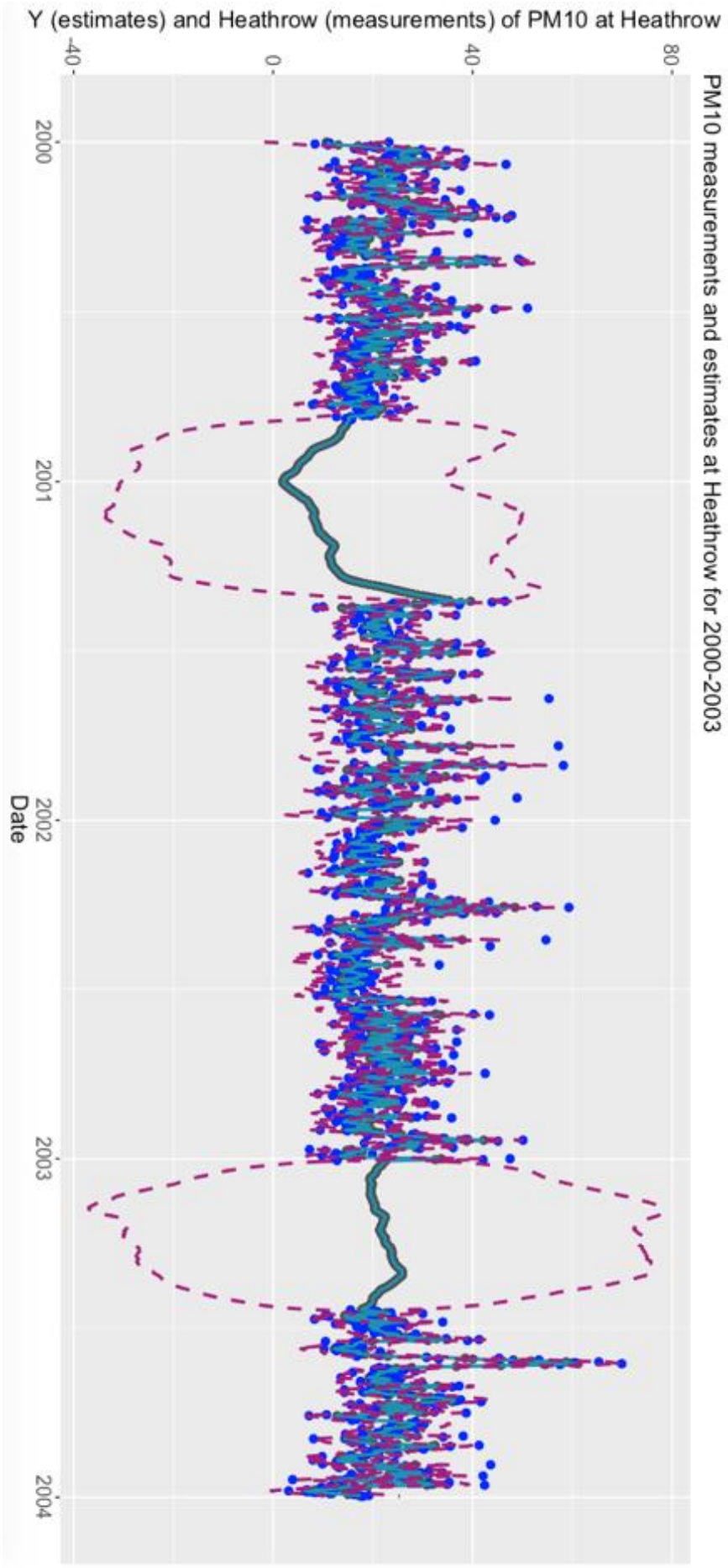


Figure 8: Line and scatter ggplot of recorded measurements ($Heathrow_t$) as well as "true" measurement parameters (Y_t) with the associated 95% credible interval boundaries shows that RW1 model approximates well to the recorded measurements ($Heathrow_t$) with tight 95% credible interval boundaries for 2000 to 2003 inclusive. However, where there are missing data intervals, the random walk model has no constraint and follows a random path until it encounters recorded measurements again, whilst the 95% credible intervals become very wide.

Second Heathrow model (jags.mod.heathrow2)

The second Heathrow model (jags.mod.heathrow2) is a random walk of order 2 (RW2) and included as output the parameters: $sigma^2_1$, $sigma^2_2$, and all Y_t. The Y_t parameters that have associated recorded measurements ($Heathrow_t$) and associated missing data both show Rhat values of > 1.1, suggesting that the chains would not converge. The density plots show that there are large differences in the estimations of $sigma^2_3$ and $sigma^2_4$ (Figures 9a and 9b) for each chain, which may relate to how each chain handled the missing data.

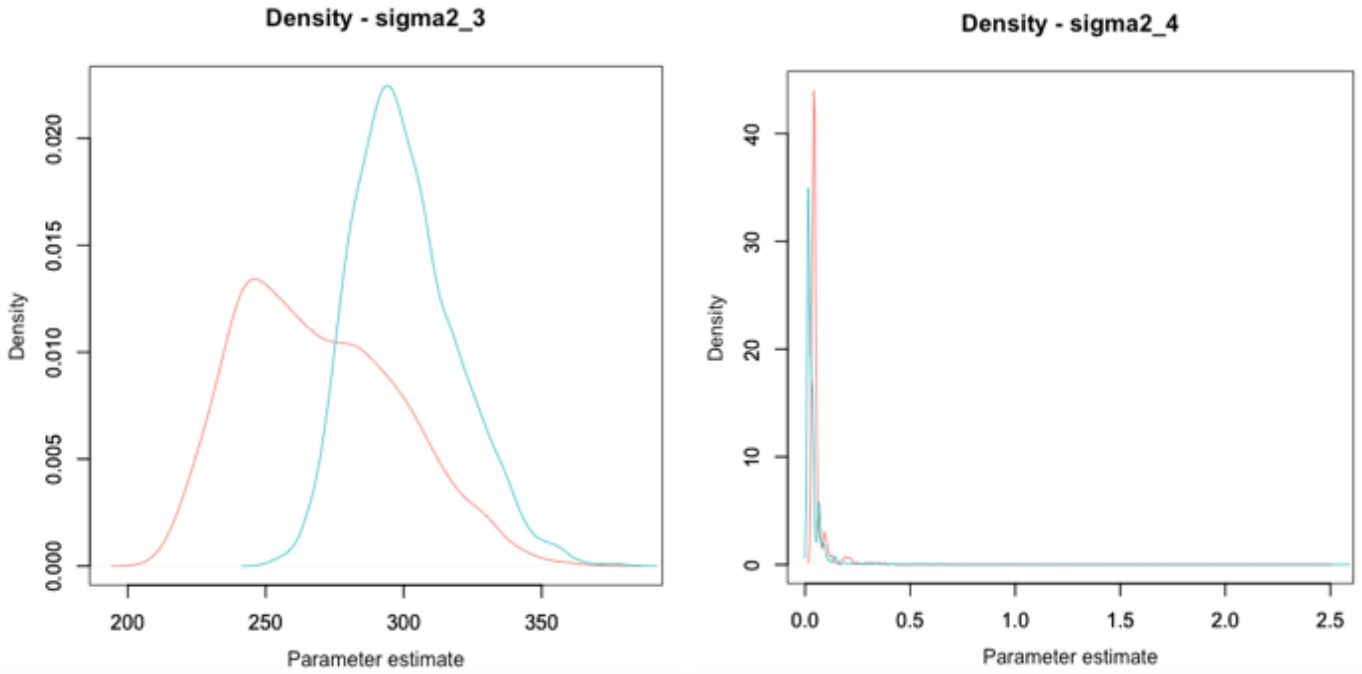
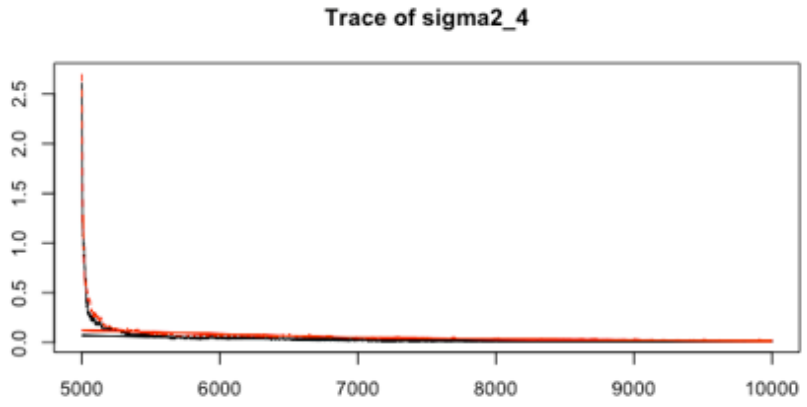
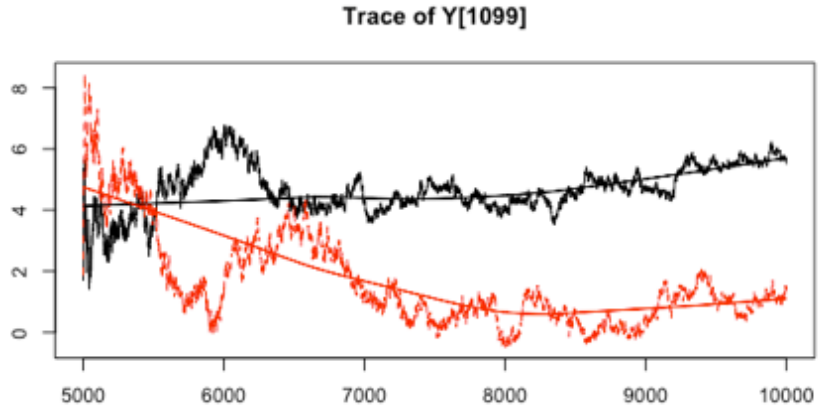


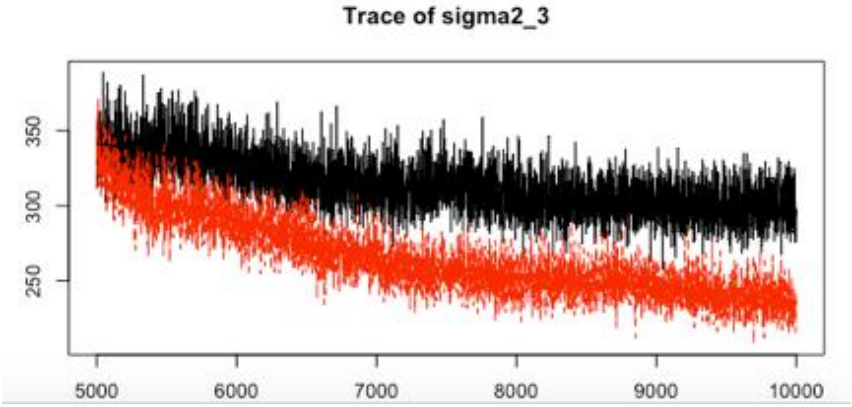
Figure 9: Density plots showing that the 2 chains have not converged for RW2 model: a) $sigma^2_3$, and b) $sigma^2_4$. This suggests that the RW2 model cause difference variances in the outputs and is not only related to the intervals of missing data.

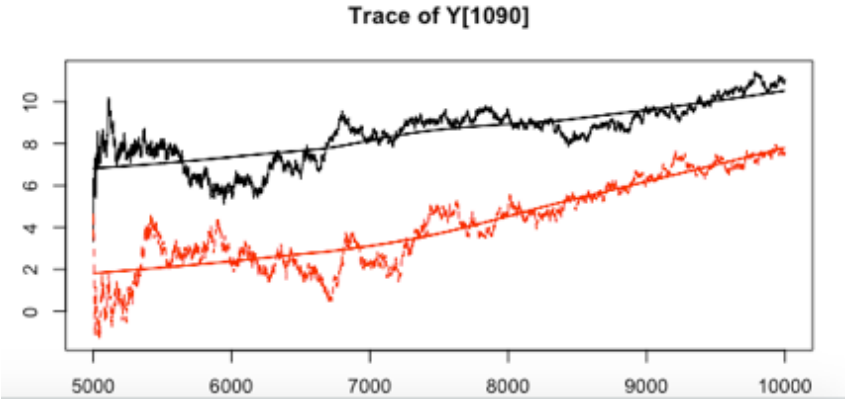
Consequently, 12 traceplots have been chosen. The first 3 show that amount of converge between the chains for deviance, $sigma^2_1$ and $sigma^2_2$ area all very poor in an RW2 model. The next 9 show how the RW2 model chains do not converge regardless of whether "true" measurement (Y_t) parameters are associated either with recorded measurements ($Heathrow_t$), namely in Y1, 1090, 1099, Y1260 and Y1265, or with the missing data, namely in Y1100, Y1110, Y1250 and Y1259 (Figure 10).

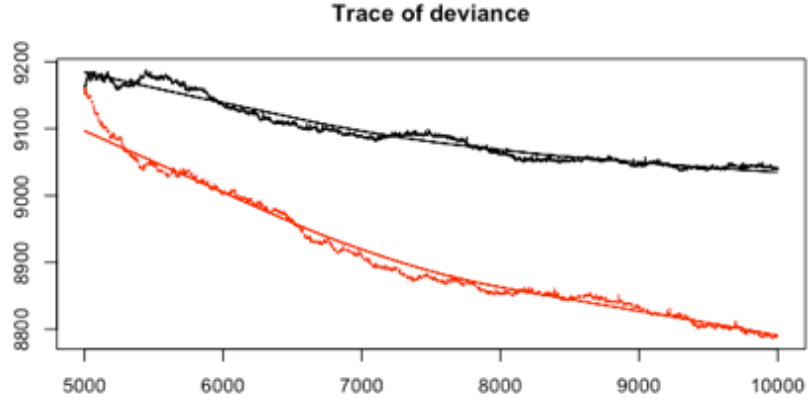
The table of the gelman diagnostics for selected parameters, namely of deviance, $sigma^2_3$, $sigma^2_4$, Y1 and Y1090 to Y1109 (Table 3) show that the point estimate and 95% CI estimate are ~1 when there are associated recorded measurements ($Heathrow_t$) and become > 1 when associated with missing data. Further, the ggplot of the recorded measurements ($Heathrow_t$) as well as the "true" measurement parameters (Y_t) with the associated 95% credible interval boundaries shows that the random walk model (RW2) does not approximate to the recorded measurements ($Heathrow_t$) and that the 95% credible interval boundaries are variable across the plot. This is shown for the first quarter of 2000 and suggests that there is no constraint on RW2 model from data or missing data alike (Figure 11). This contrasts with results from the RW1 model for the same period of time (Figure 12).

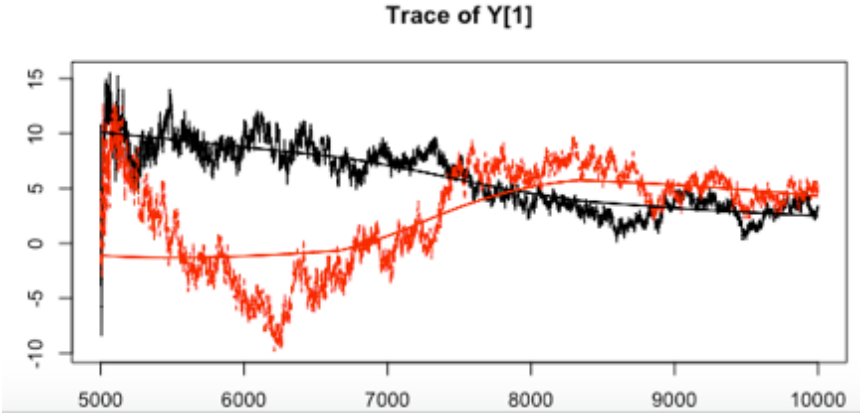

Trace of sigma2_4
0.0
0.5
1.0
1.5
2.0
2.5
5000
6000
7000
8000
9000
10000


Trace of Y[1099]
0
2
4
6
8
5000
6000
7000
8000
9000
10000


Trace of sigma2_3
250
300
350
5000
6000
7000
8000
9000
10000


Trace of Y[1090]
0
2
4
6
8
10
5000
6000
7000
8000
9000
10000


Trace of deviance
8800
8900
9000
9100
9200
5000
6000
7000
8000
9000
10000


Trace of Y[1]
-10
-5
0
5
10
15
5000
6000
7000
8000
9000
10000

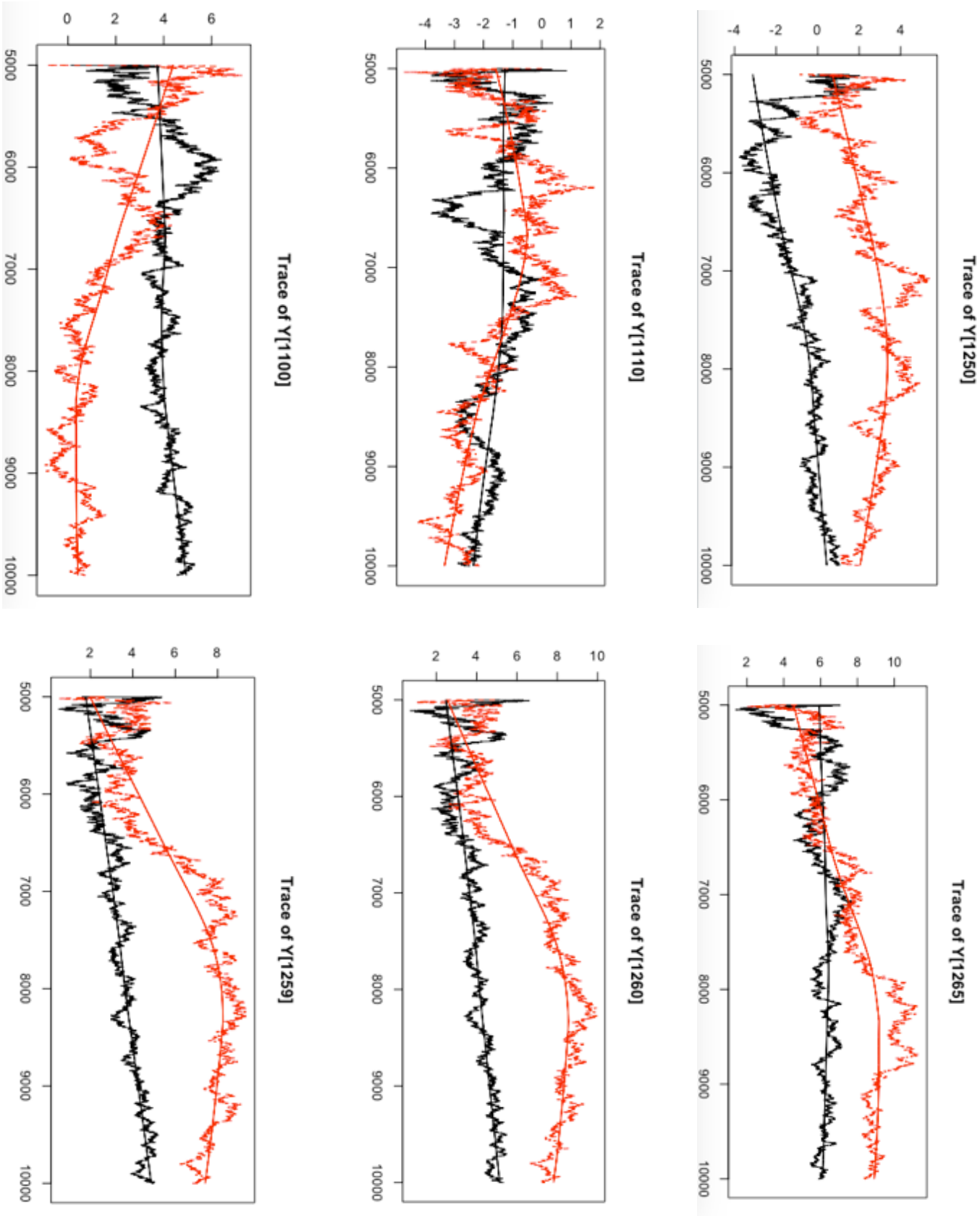


Figure 10: Progression of 12 traceplots showing how the 2 chains do not converge well for deviance, $sigma^2_3$ and $sigma^2_4$ are in an RW2 model. The next 9 show how the RW2 model chains do not converge regardless of whether "true" measurement (Y_t) parameters are associated either with recorded measurements ($Heathrow_t$), namely in Y1, 1090, 1099, Y1260 and Y1265, or with the missing data, namely in Y1100, Y1110, Y1250 and Y1259.

	Point est.	Upper C.I.
deviance	2.094714	5.569458
sigma2_3	1.808432	3.612728
sigma2_4	1.018440	1.084706
Y[1]	1.747732	3.094974
Y[1090]	1.310199	2.086784
Y[1091]	1.584568	2.831164
Y[1092]	1.961390	3.700543
Y[1093]	2.388124	4.685944
Y[1094]	2.739002	5.793981
Y[1095]	2.896588	6.918596
Y[1096]	2.839649	7.570409
Y[1097]	2.648993	7.606738
Y[1098]	2.413135	7.411857
Y[1099]	2.113467	6.589198
Y[1100]	1.710023	4.679388
Y[1101]	1.319917	2.478988
Y[1102]	1.132433	1.157433
Y[1103]	1.183718	1.655968
Y[1104]	1.634476	3.018707
Y[1105]	2.237607	4.560091
Y[1106]	2.792888	5.906080
Y[1107]	3.255175	6.954553
Y[1108]	3.630744	7.735158
Y[1109]	3.920556	8.348367

Table 3: Results table of the gelman diagnostics for selected parameters from RW2 model, namely of deviance, $sigma^2_3$, $sigma^2_4$, Y1 and Y1090 to Y1109 show that the point estimate and 95% CI estimate are > 1 whether they are associated either with recorded measurements ($Heathrow_t$) or with missing data

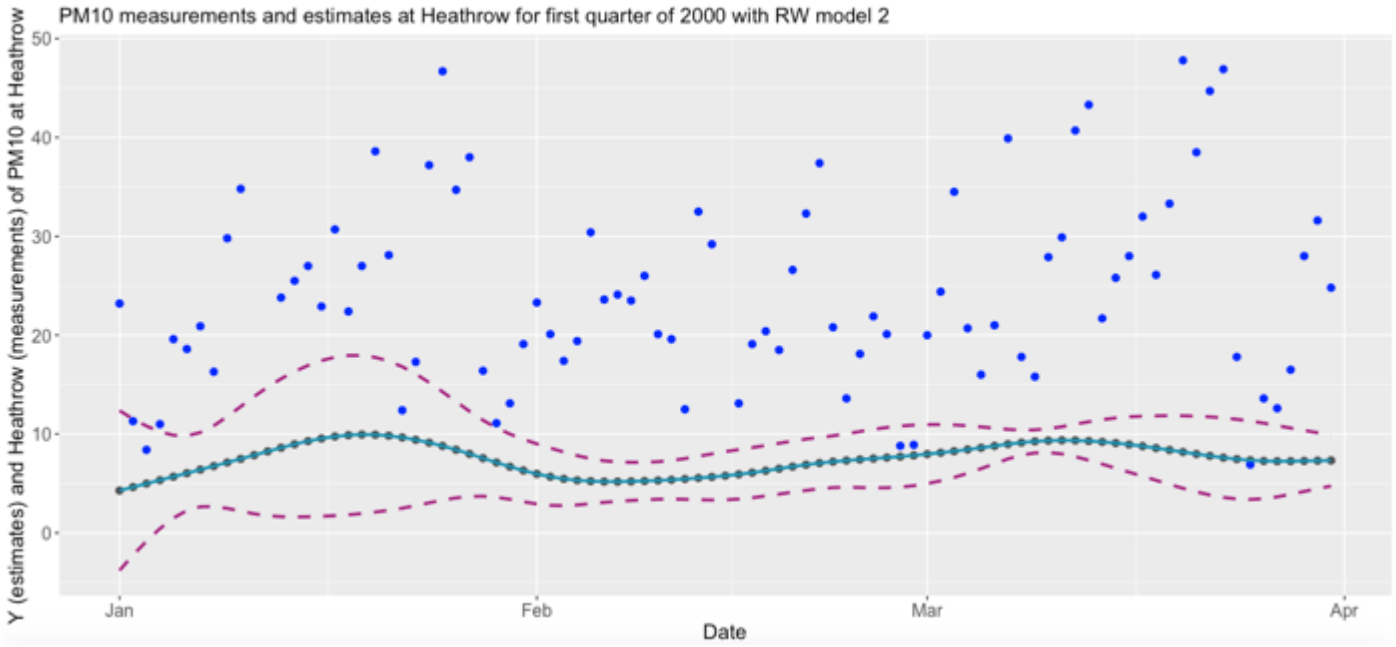


Figure 11: Line and scatter ggplot of the recorded measurements ($Heathrow_t$) as well as the "true" measurement parameters (Y_t) with the associated 95% credible interval boundaries shows that RW2 model does not approximate well to the recorded measurements ($Heathrow_t$) and that the 95% credible interval boundaries are variable across the plot. This is shown for the first quarter of 2000 and suggests that there is no constraint on RW2 model from data or missing data alike

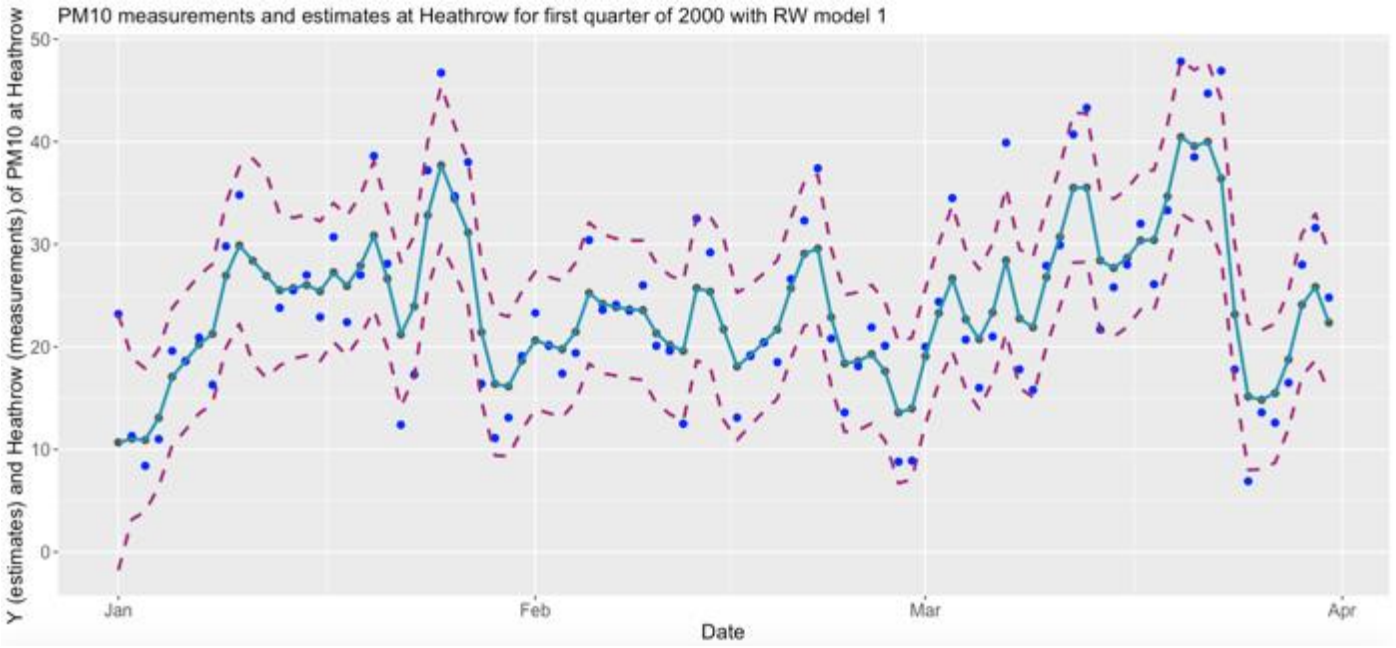


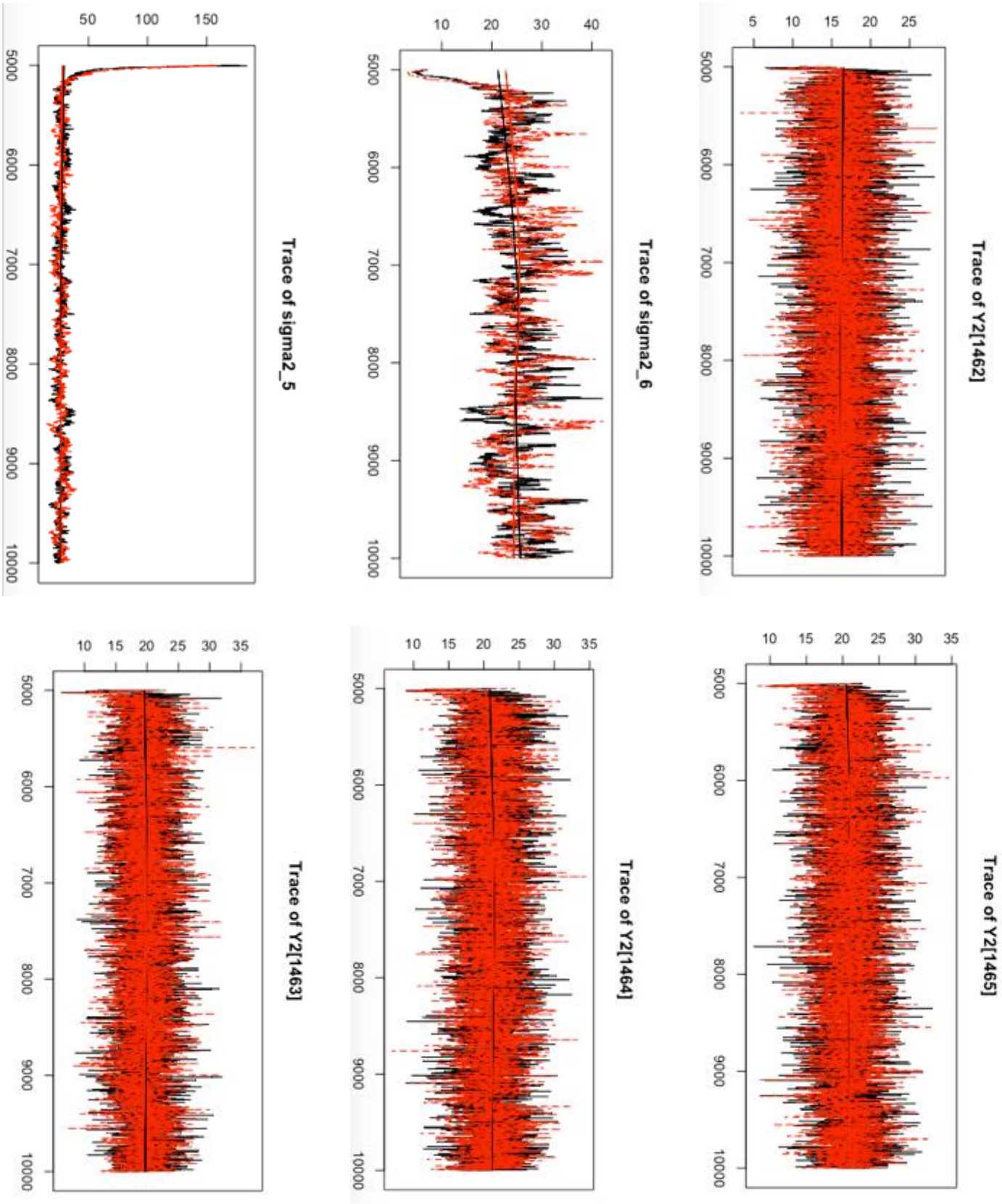
Figure 12: Line and scatter ggplot of recorded measurements ($Heathrow_t$) as well as "true" measurement parameters (Y_t) with the associated 95% credible interval boundaries shows that RW1 model approximates well to the recorded measurements ($Heathrow_t$) with tight 95% credible interval boundaries for the first quarter of 2000.

Heathrow RW1 (jags.mod.heathrow3) model to forecast data

The third Heathrow model (jags.mod.heathrow3) is a random walk of order 1 (RW1) and the fourth Heathrow model (jags.mod.heathrow4) is a random walk of order 2 (RW2). Both models include as output the parameters: $sigma^2_1$, $sigma^2_2$, and all $Y2_t$.

As with the first Heathrow model, the $Y2_t$ parameters for the third Heathrow model have associated recorded measurements ($Heathrow_t$) that show Rhat values of ~ 1, suggesting that the chains would converge, whilst the $Y2_t$ parameters that have associated missing data show Rhat values > 1.1, suggesting that they would not converge. The density plots show that there are some minor differences in the estimations of $sigma^2_5$ and $sigma^2_6$ compared to $sigma^2_1$ and $sigma^2_2$ (see figures 6a ad 6b) for each chain, which may relate to how each chain handled the missing data. However, there are now 7 additional Y_t parameters to estimate that are associated with data that was

deliberately removed for the first week in 2004. Consequently, 9 traceplots have been chosen. The first 2 show that amount of converge between the chains for $sigma^2_1$ and $sigma^2_2$ area poor in an RW1 model. The next 7 traceplots show how the RW1 model chains converge for the forecasted measurements (Y1462-Y1468) although there is missing data (Figure 13). The table of the gelman diagnostics for selected parameters, namely of $sigma^2_5$, $sigma^2_6$, and Y2 1462 to Y2 1468 (Table 4) show that the point estimate and 95% CI estimate are ~1 whether they are associated either with recorded measurements ($Heathrow_t$) or with missing data.



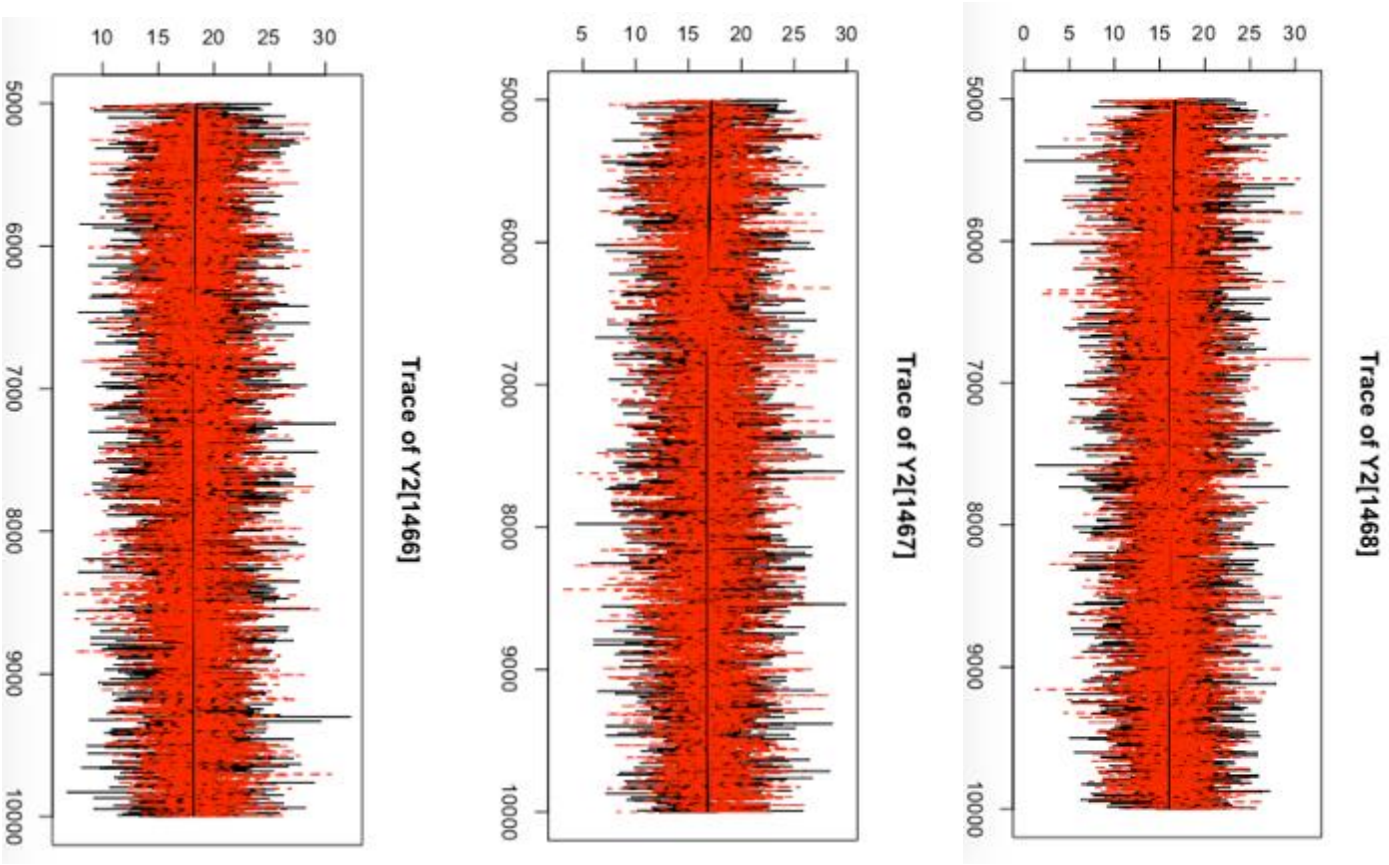


Figure 13: Progression of 9 traceplots showing how the 2 chains do not converge well for $sigma^2_5$ and $sigma^2_6$ in an RW1 model. The next 7 traceplots are the $Y2_t$ estimates in the last week of 2004 showing how the 2 chains continue to converge in RW1 model despite the recorded measurements ($Heathrow_t$) being deliberately missing

	Point est.	Upper C.I.
deviance	1.0076707	1.0269753
sigma2_5	1.0013016	1.0068836
sigma2_6	1.0056469	1.0182560
Y2[1461]	1.0001132	1.0003916
Y2[1462]	1.0000044	1.0004211
Y2[1463]	1.0000118	1.0000120
Y2[1464]	1.0000619	1.0001247
Y2[1465]	0.9999500	1.0000333
Y2[1466]	1.0000400	1.0000555
Y2[1467]	1.0003258	1.0005184
Y2[1468]	1.0001313	1.0003285

Table 4: Results table of the gelman diagnostics for selected parameters, namely of $sigma^2_5$, $sigma^2_6$, and Y2 1462 to Y2 1468 show that the point estimate and 95% CI estimate are ~1 whether they are associated either with recorded measurements ($Heathrow_t$) or with missing data.

Further, the ggplot of the recorded measurements ($Heathrow_t$) as well as the "true" measurement parameters ($Y2_t$) with the associated 95% credible interval boundaries shows that the random walk model approximates well to the recorded measurements ($Heathrow_t$) with tight 95% credible interval boundaries for the last quarter of 2003 and the first week of 2004. In this first week of 2004 with the missing data, the random walk model has no constraint and follows a straight line, whilst the 95% credible intervals become wider (Figure 14).

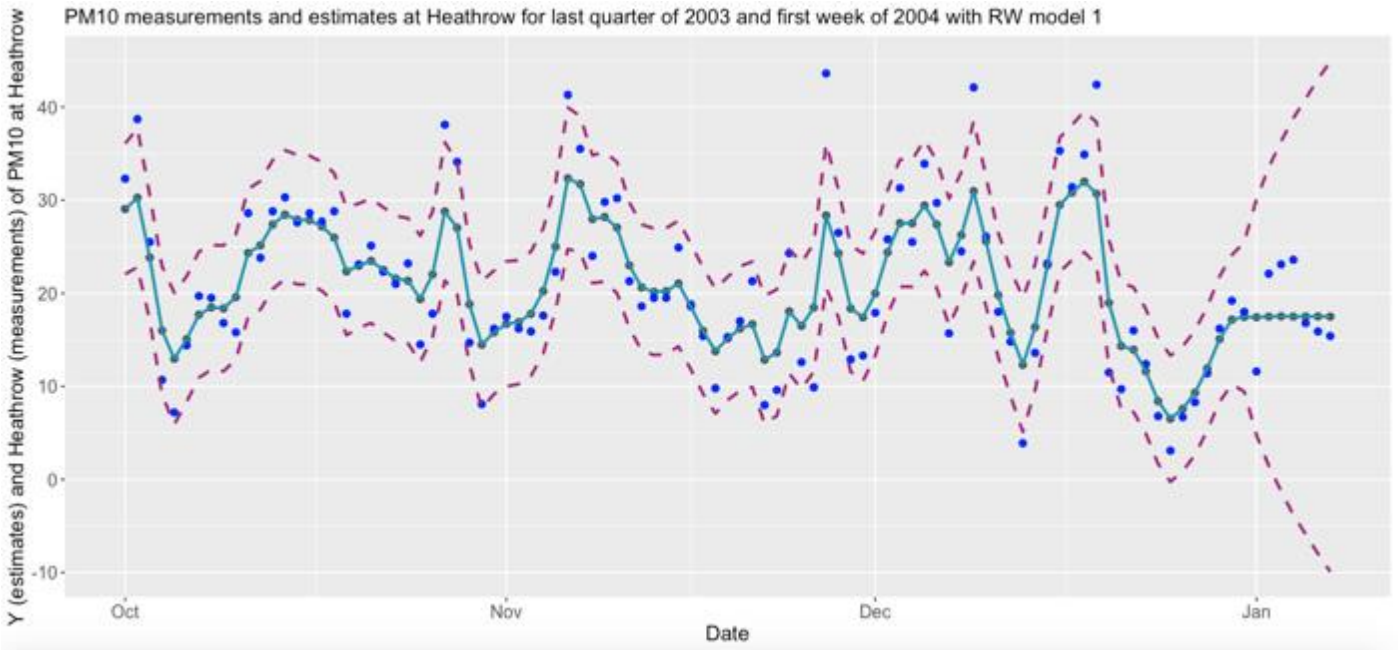


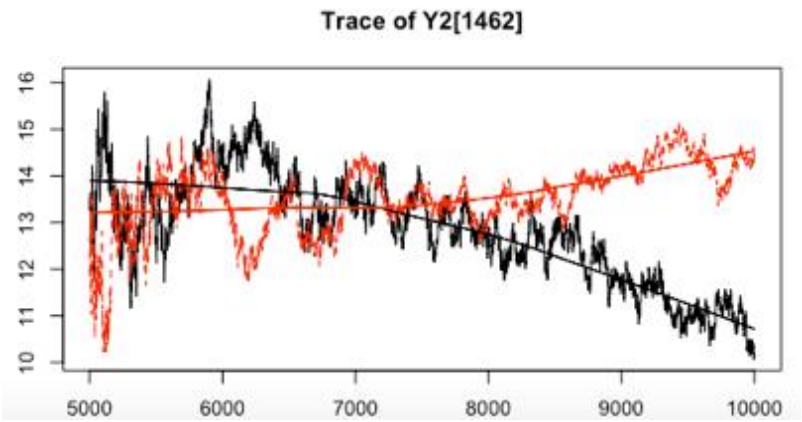
Figure 14: Line and scatter ggplot of the recorded measurements ($Heathrow_t$) as well as the "true" measurement parameters ($Y2_t$) with the associated 95% credible interval boundaries shows that RW1 model approximates well to the recorded measurements ($Heathrow_t$) with tight 95% credible interval boundaries for the last quarter of 2003 and the first week of 2004. In this first week of 2004 with the missing data, the random walk model has no constraint and follows a straight line, whilst the 95% credible intervals become wider

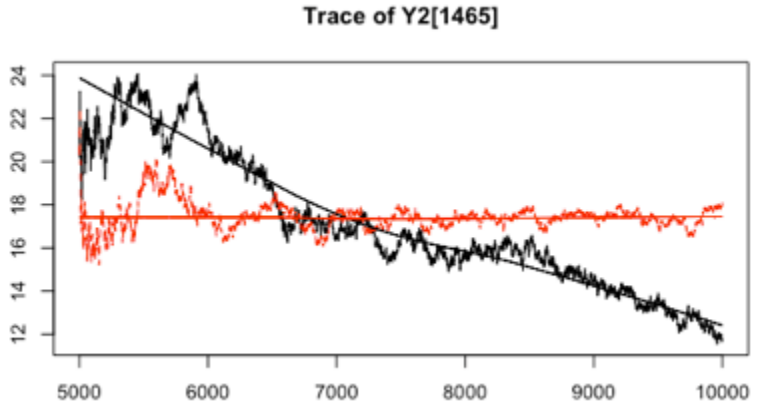
Heathrow RW2 (jags.mod.heathrow4) model to forecast data

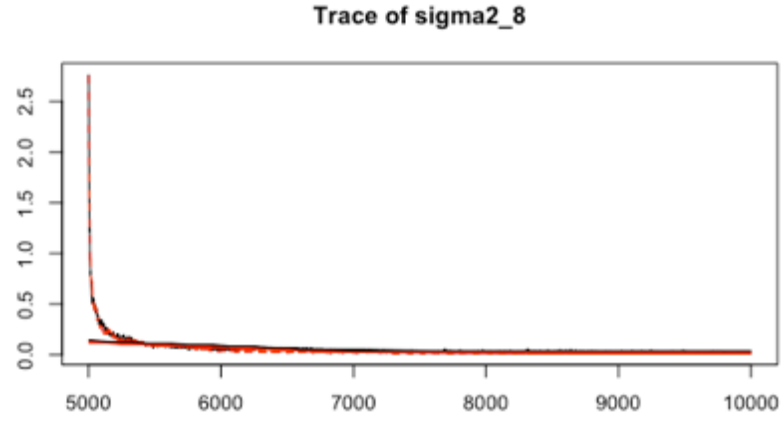
As with the second Heathrow model, the $Y2_t$ parameters for the fourth Heathrow model have associated recorded measurements ($Heathrow_t$) and associated missing data both show Rhat values of > 1.1, suggesting that the chains would not converge. The density plots show that there are large differences in the estimations of $sigma^2_7$ and $sigma^2_8$ compared to $sigma^2_3$ and $sigma^2_4$ (see figures 9a and 9b) for each chain, which relate to how each chain handled the data. However, there are now 7 additional $Y2_t$ parameters to estimate that are associated with data that was deliberately removed for the first week in 2004. Consequently, 9 traceplots have been chosen, with the $sigma^2_1$, $sigma^2_2$ and forecasted measurements (Y1462-Y1468) all showing that the RW2 model chains do not converge (Figure 15).

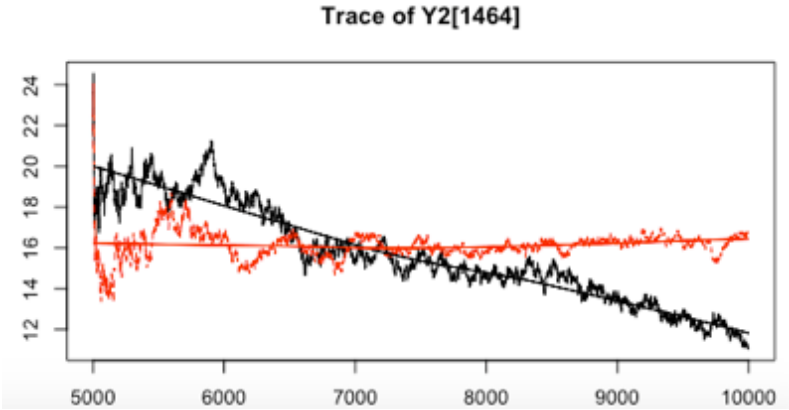
The table of the gelman diagnostics for selected parameters, namely of $sigma^2_7$, $sigma^2_8$, and Y2 1462 to Y2 1468 (Table 5) show that the point estimate and 95% CI estimate are > 1 whether they are associated either with recorded measurements ($Heathrow_t$) or with missing data.

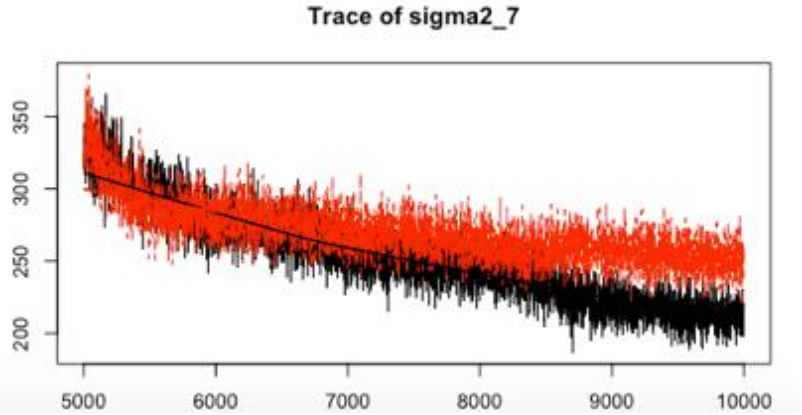
Further, the ggplot of the recorded measurements ($Heathrow_t$) as well as the "true" measurement parameters ($Y2_t$) with the associated 95% credible interval boundaries shows that the random walk model does not approximate to the recorded measurements ($Heathrow_t$) and that the 95% credible interval boundaries are very wide across the plot. This is shown for the last quarter of 2003 and first week of 2004 and suggests that there is no constraint on RW2 model from data or missing data alike. However, in the last week of 2004, although there is no variation in the random path of RW2 model, the 95% credible intervals become wider (Figure 16).

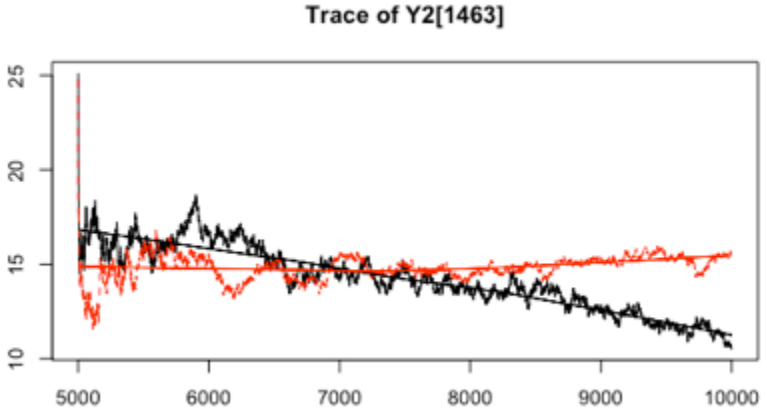

Trace of Y2[1462]
10
11
12
13
14
15
16
5000
6000
7000
8000
9000
10000


Trace of Y2[1465]
12
14
16
18
20
22
24
5000
6000
7000
8000
9000
10000


Trace of sigma2_8
0.0
0.5
1.0
1.5
2.0
2.5
5000
6000
7000
8000
9000
10000


Trace of Y2[1464]
12
14
16
18
20
22
24
5000
6000
7000
8000
9000
10000


Trace of sigma2_7
200
250
300
350
5000
6000
7000
8000
9000
10000


Trace of Y2[1463]
10
15
20
25
5000
6000
7000
8000
9000
10000

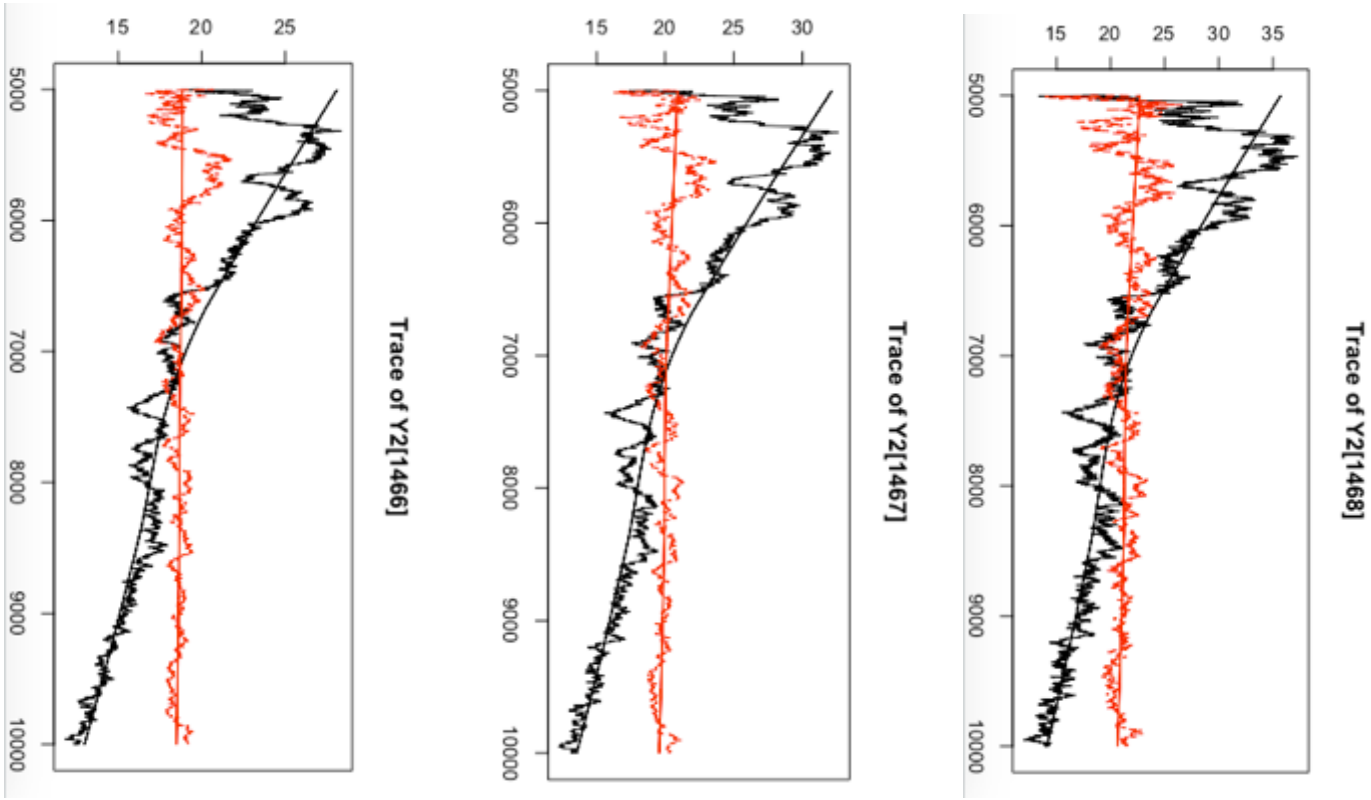


Figure 15: Progression of 6 traceplots from the $Y2_t$ estimates using RW2 in the last week of 2004 showing how the 2 chains do not converge with recorded measurements ($Heathrow_t$) being deliberately missing

	Point est.	Upper C.I.
deviance	1.425745	2.372184
sigma2_7	1.216058	1.721989
sigma2_8	1.000573	1.003197
Y2[1461]	1.879871	3.589926
Y2[1462]	1.714941	3.185679
Y2[1463]	1.593297	2.868584
Y2[1464]	1.502397	2.619745
Y2[1465]	1.434163	2.426380
Y2[1466]	1.380243	2.267197
Y2[1467]	1.337345	2.136874
Y2[1468]	1.302696	2.029363

Table 5: Results table of the gelman diagnostics for selected parameters, namely of $sigma^2_7$, $sigma^2_8$, and Y2 1462 to Y2 1468 show that the point estimate and 95% CI estimate are > 1 whether they are associated either with recorded measurements ($Heathrow_t$) or with missing data.

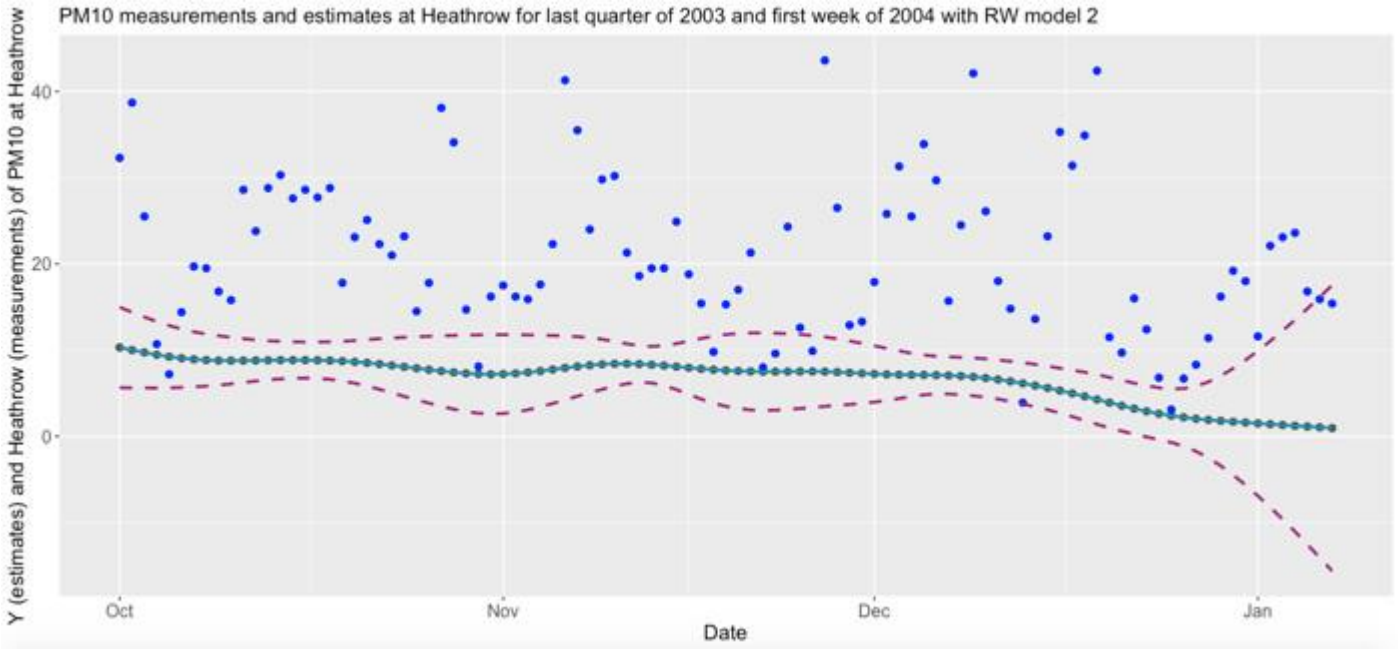


Figure 16: Line and scatter ggplot of the recorded measurements ($Heathrow_t$) as well as the "true" measurement parameters ($Y2_t$) with the associated 95% credible interval boundaries shows that the random walk model does not approximate to the recorded measurements ($Heathrow_t$) and that the 95% credible interval boundaries are very wide across the plot. This is shown for the last quarter of 2003 and first week of 2004 and suggests that there is no constraint on RW2 model from data or missing data alike. However, in the last week of 2004, although there is no variation in the random path of RW2 model, the 95% credible intervals become wider

Root mean square error for Heathrow RW1 (jags.mod.heathrow7) model

The fifth Heathrow model (jags.mod.heathrow7) is a random walk of order 1 (RW1) and included as output the parameters: $sigma^2_{21}$, $sigma^2_{22}$, all Y_t and all rmse.Y_t. The rmse.Y_t parameters that have associated recorded measurements ($Heathrow_t$) show Rhat values of ~ 1, suggesting that the chains would converge, whilst the Y_t parameters that have associated missing data show Rhat values > 1.1, suggesting that they would not converge. The density plots show that there are some minor differences in the estimations of $sigma^2_{21}$ and $sigma^2_{22}$ (figures 17a and 17b) for each chain, which may relate to how each chain handled the missing data.

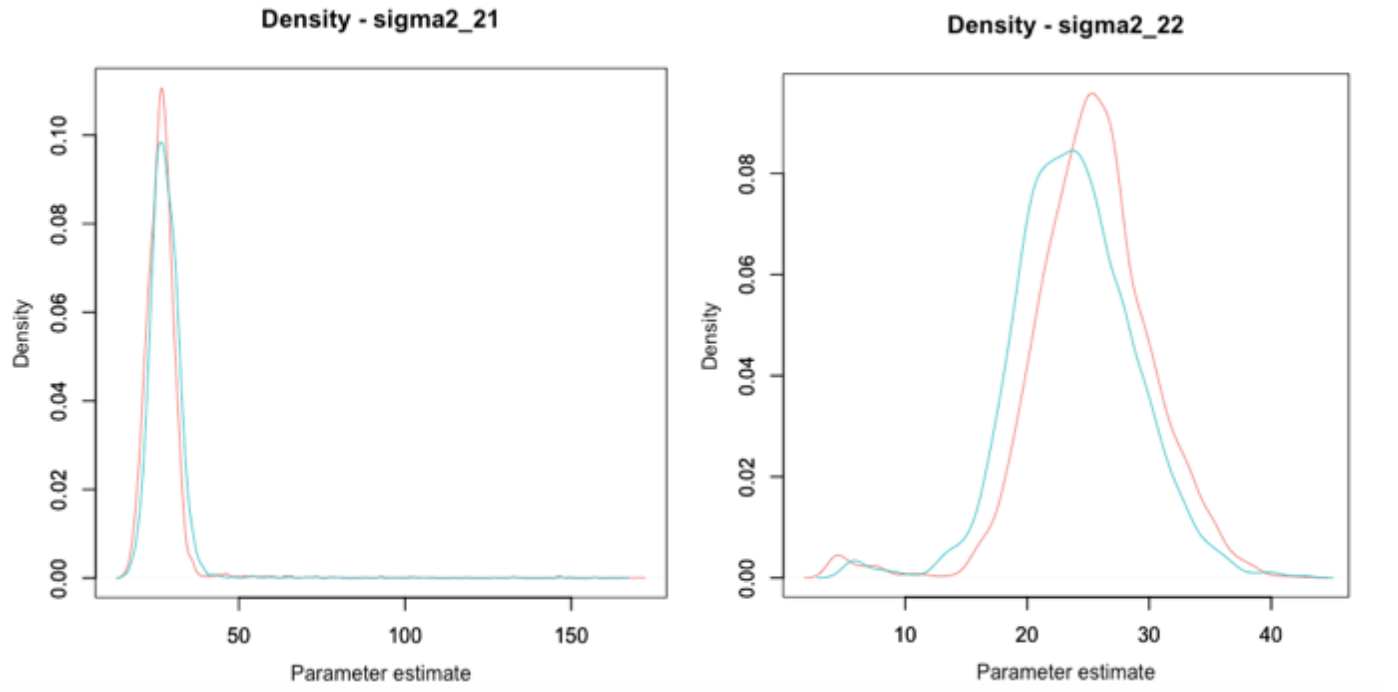
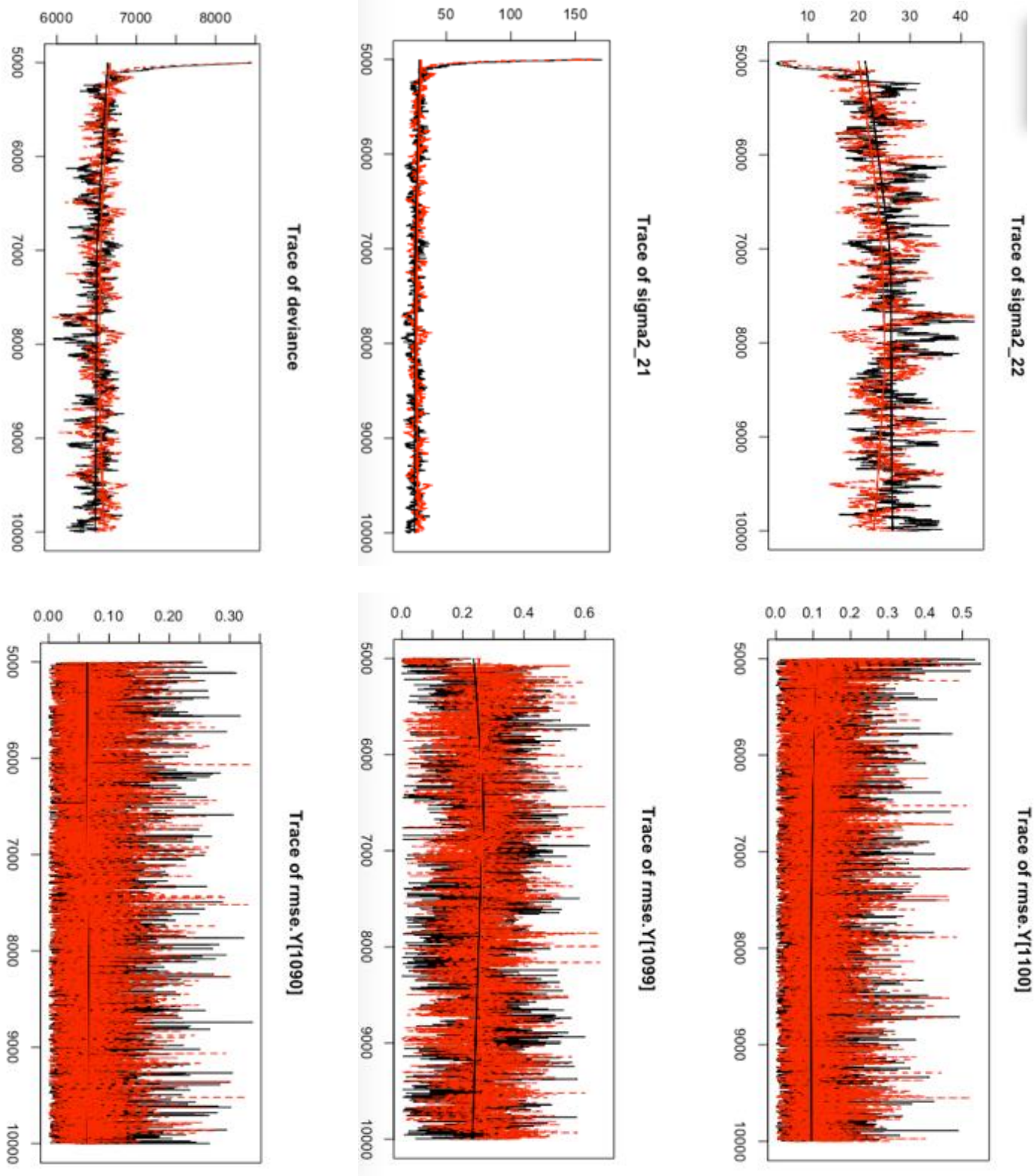


Figure 17: Density plots showing that the 2 chains have not quite converged for RW1 model: a) $sigma^2_{21}$, and b) $sigma^2_{22}$. This suggests that parts of the data cause difference variances in the outputs, which may result from the intervals of missing data.

Consequently, 11 traceplots have been chosen. The first 3 show that amount of converge between the chains for deviance, $sigma^2_{21}$ and $sigma^2_{22}$ area all poor in an RW1 model. The next 8 show how the RW1 model chains converge whether "true" measurement (Y_t) parameters either have associated recorded measurements ($Heathrow_t$), namely rmse.Y1090, rmse.Y1099, rmse.Y1260 and rmse.Y1265, or have associated missing data, namely rmse.Y1100, rmse.Y1110, rmse.Y1250 and rmse.Y1259 (Figure 18).



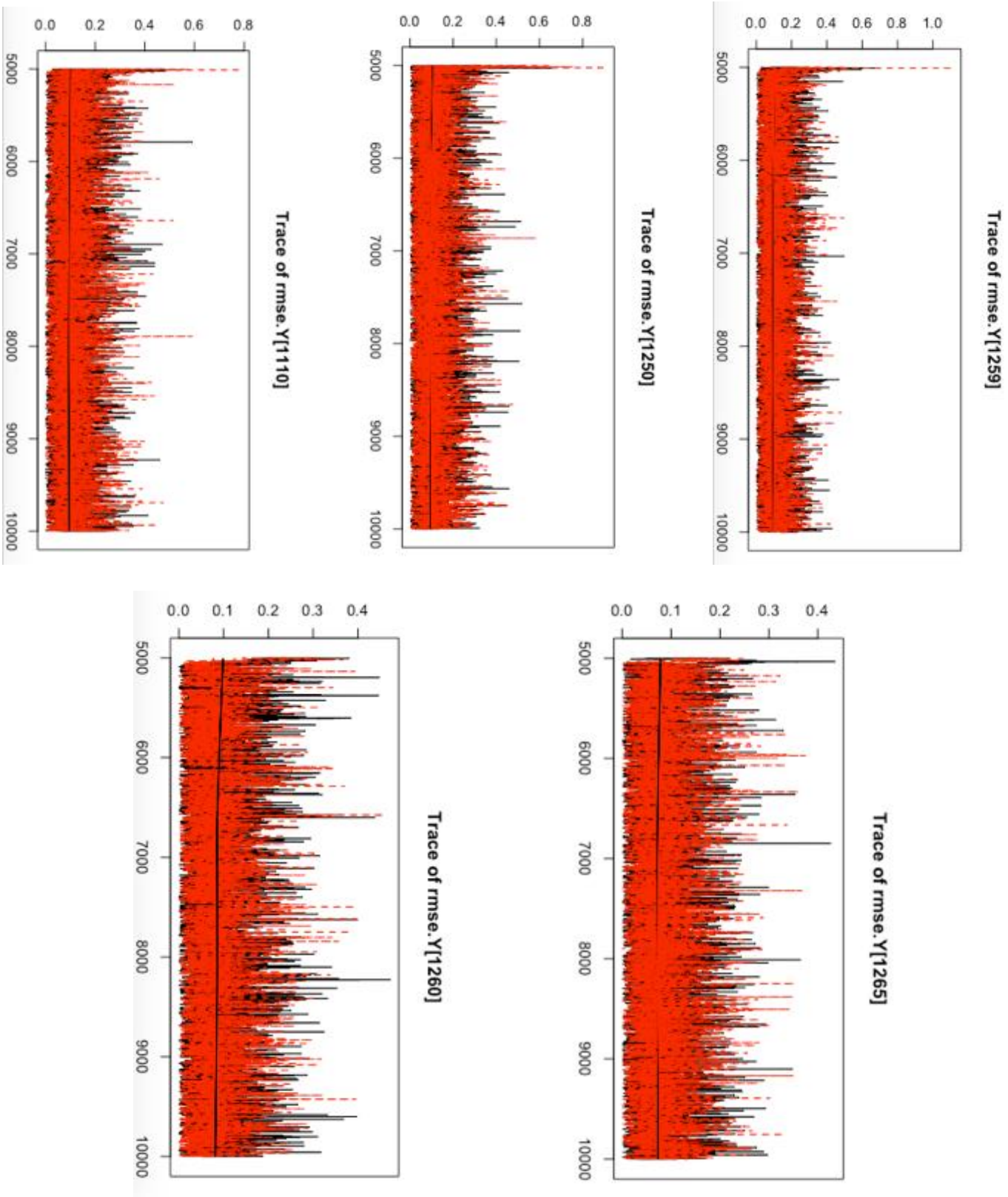


Figure 18: Progression of 11 traceplots showing how the 2 chains do not converge well for deviance, $sigma^2_{21}$ and $sigma^2_{22}$ in an RW1 model. The next 8 show how the RW1 model chains converge whether "true" measurement (Y_t) parameters either have associated recorded measurements ($Heathrow_t$), namely rmse.Y1090, rmse.Y1099, rmse.Y1260 and rmse.Y1265, or missing data, namely rmse.Y1100, rmse.Y1110, rmse.Y1250 and rmse.Y1259.

The table of the gelman diagnostics for selected parameters, namely of deviance, $sigma^2_{21}$, $sigma^2_{22}$ and rmse.Y1090 to rmse.Y1110 (Table 6) show that the point estimate and 95% CI estimate are ~1 whether they are associated either with recorded measurements ($Heathrow_t$) or with missing data. Further, the ggplot of root mean squared error estimates ($rmse.Y_t$) with associated 95% credible interval boundaries showing that RW1 model has generally low RMSE with tight 95% credible interval boundaries for 2000 to 2003 inclusive. Where there are missing data intervals, the random walk model provides a consistent RMSE value of ~0.1 and consistent 95% credible intervals (upper ~0.28; lower ~-0.06) (Figure 19).

	Point est.	Upper C.I.
deviance	1.0306764	1.1003581
sigma2_21	1.0238362	1.0398680
sigma2_22	1.0566784	1.2120995
rmse.Y[1090]	1.0002404	1.0003640
rmse.Y[1091]	0.9999158	0.9999201
rmse.Y[1092]	1.0005430	1.0015560
rmse.Y[1093]	1.0002025	1.0013761
rmse.Y[1094]	1.0052182	1.0231194
rmse.Y[1095]	1.0034803	1.0137054
rmse.Y[1096]	1.0007283	1.0040251
rmse.Y[1097]	1.0177038	1.0233705
rmse.Y[1098]	1.0173650	1.0190804
rmse.Y[1099]	1.0035288	1.0171433
rmse.Y[1100]	1.0001406	1.0007324
rmse.Y[1101]	1.0001726	1.0004543
rmse.Y[1102]	1.0010496	1.0032914
rmse.Y[1103]	0.9999306	1.0000303
rmse.Y[1104]	1.0009487	1.0030516
rmse.Y[1105]	1.0008765	1.0040085
rmse.Y[1106]	1.0001939	1.0002310
rmse.Y[1107]	1.0008863	1.0031777
rmse.Y[1108]	0.9999850	1.0003080
rmse.Y[1109]	1.0000586	1.0006927

Table 6: Results table of the gelman diagnostics for selected parameters for RW1 model, namely of deviance, $sigma^2_{21}$, $sigma^2_{22}$, rmse.Y1090 to rmse.Y1110 show that the point estimate and 95% CI estimate are ~1 whether there are associated recorded measurements ($Heathrow_t$) or missing data

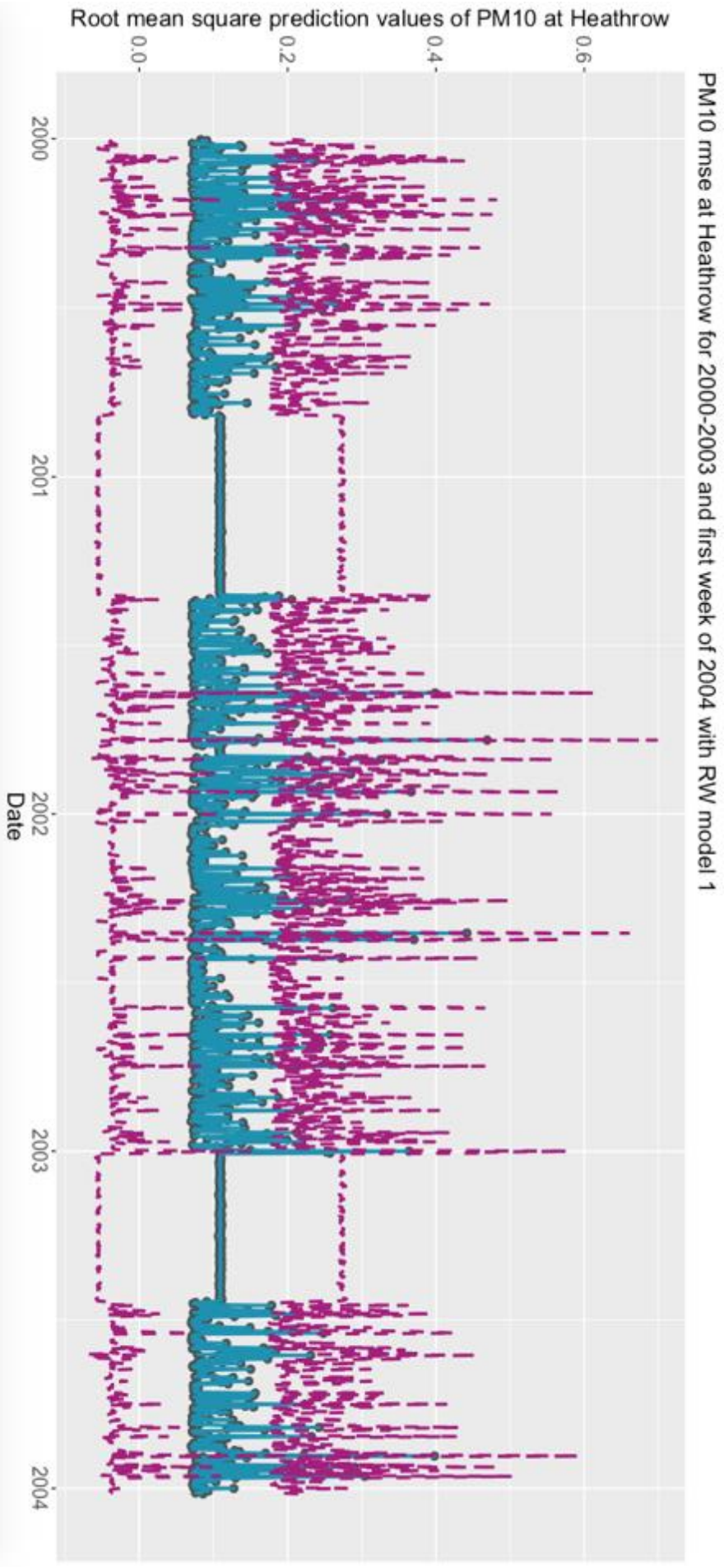


Figure 19: Line and scatter ggplot of root mean squared error estimates (rmse.Y_t) with associated 95% credible interval boundaries showing that RW1 model has generally low RMSE with tight 95% credible interval boundaries for 2000 to 2003 inclusive. Where there are missing data intervals, the random walk model provides a consistent RMSE value of ~0.1 and consistent 95% credible intervals (upper ~0.28; lower ~-0.06).

Root mean square error for Heathrow RW2 (jags.mod.heathrow8) model

The sixth Heathrow model (jags.mod.heathrow8) is a random walk of order 2 (RW1) and included as output the parameters: $sigma^2_{23}$, $sigma^2_{24}$, all Y_t and all $rmse.Y_t$. The $rmse.Y_t$ parameters that have associated recorded measurements ($Heathrow_t$) and missing data both show Rhat values of > 1, suggesting that the chains would not converge. The density plots show that there are large differences in the estimations of $sigma^2_{23}$ and $sigma^2_{24}$ (figures 20a and 20b) for each chain, which may relate to how each chain handled the missing data.

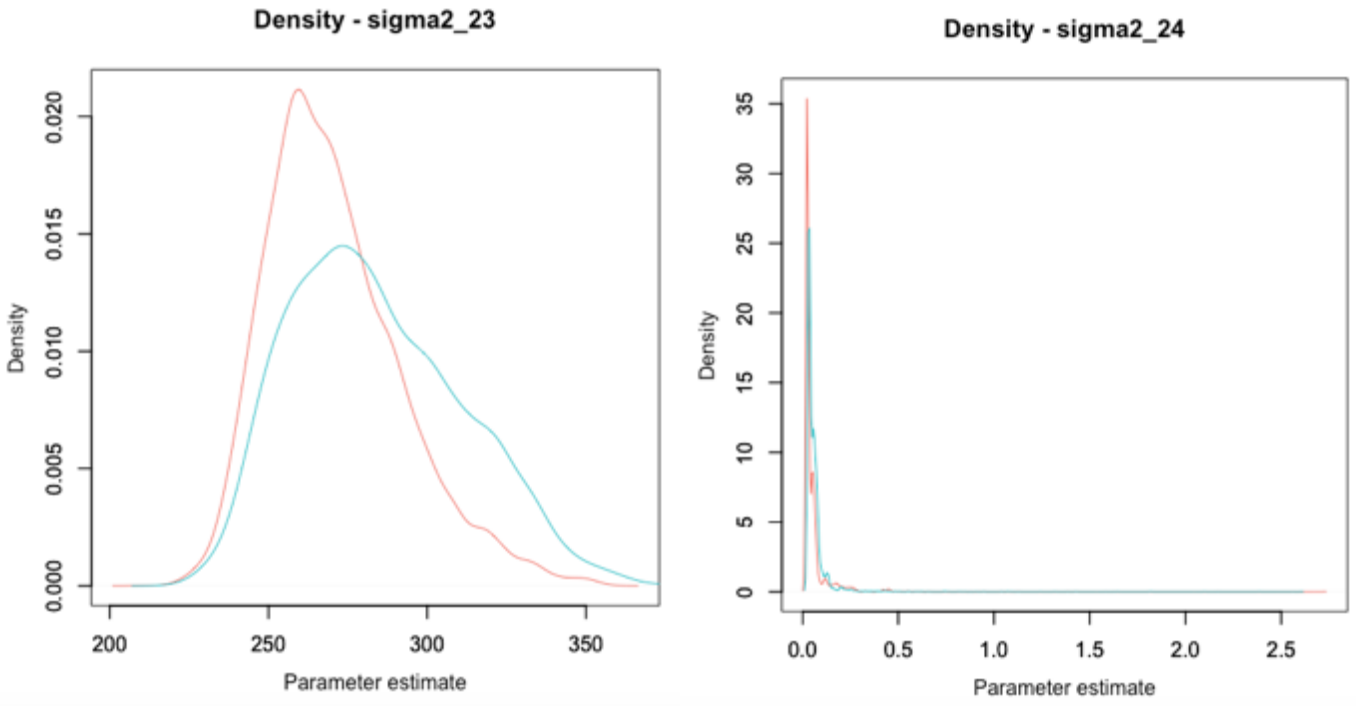
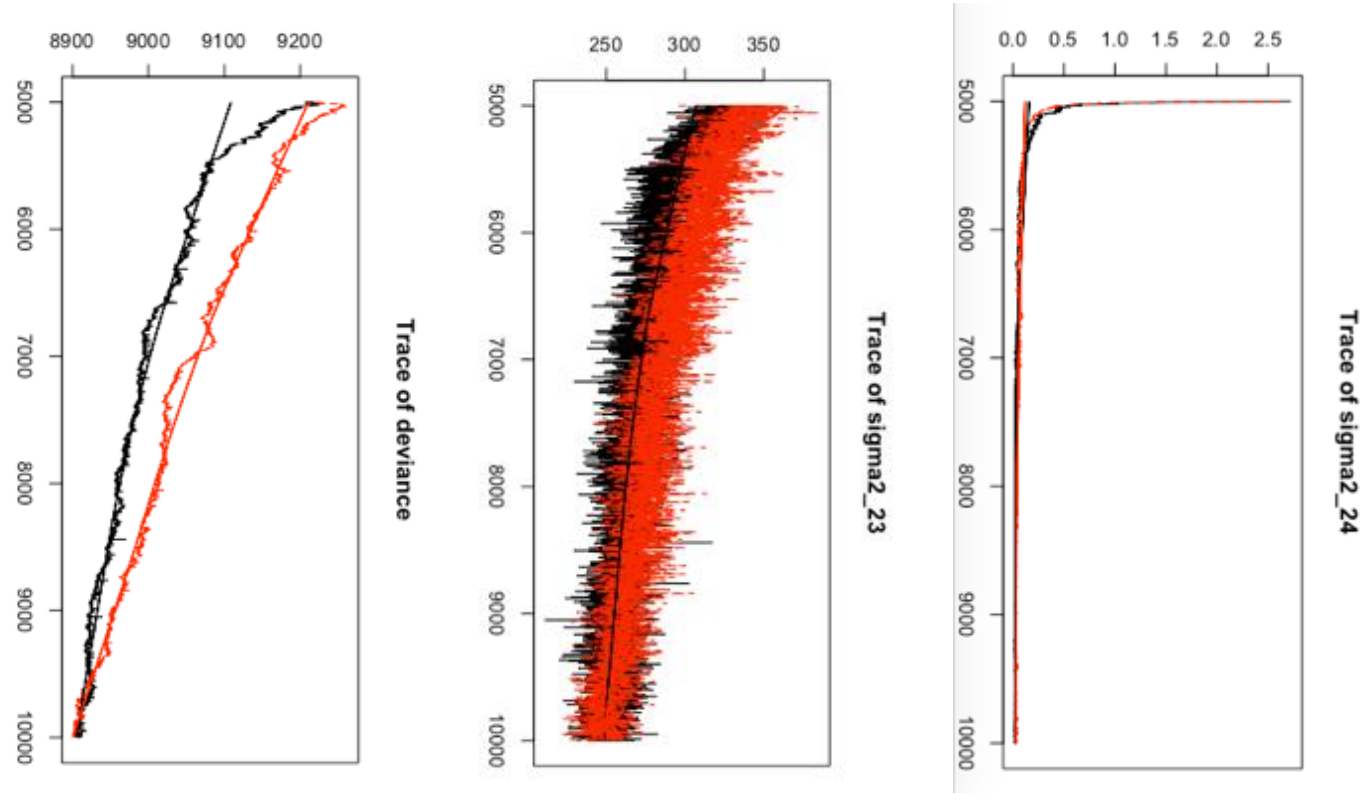
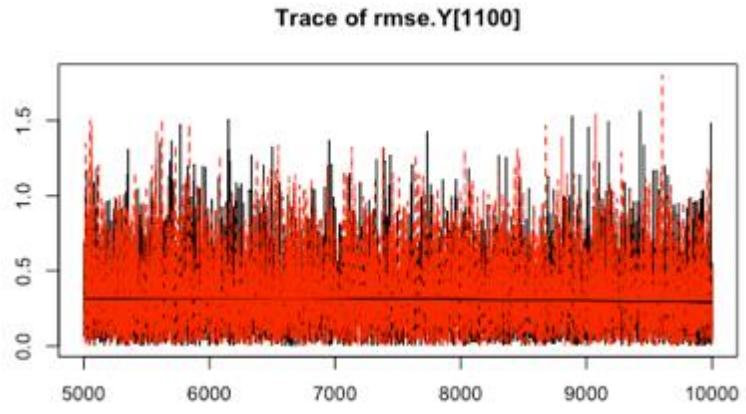
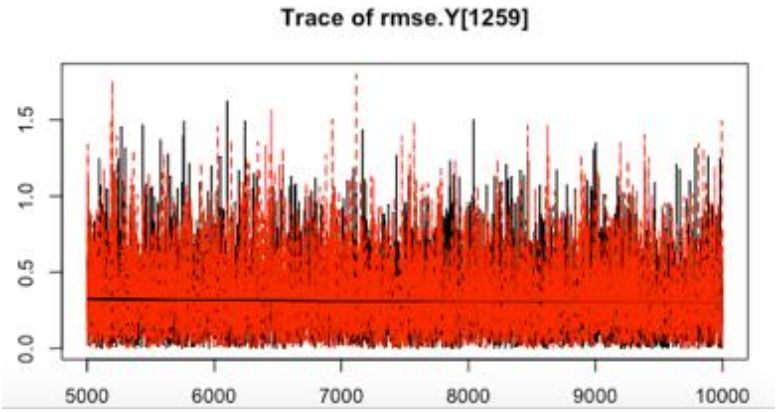
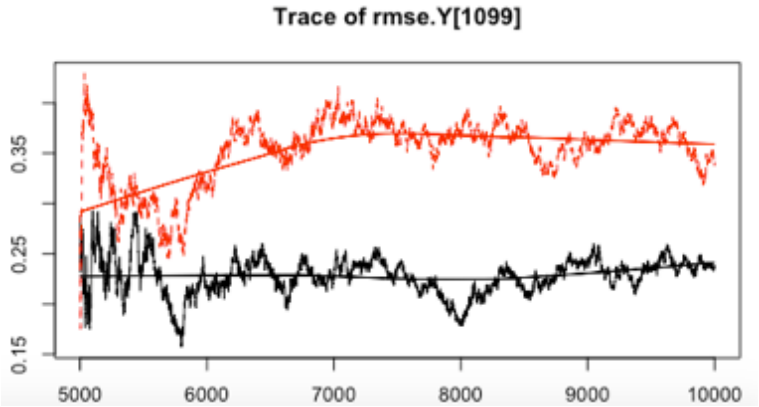


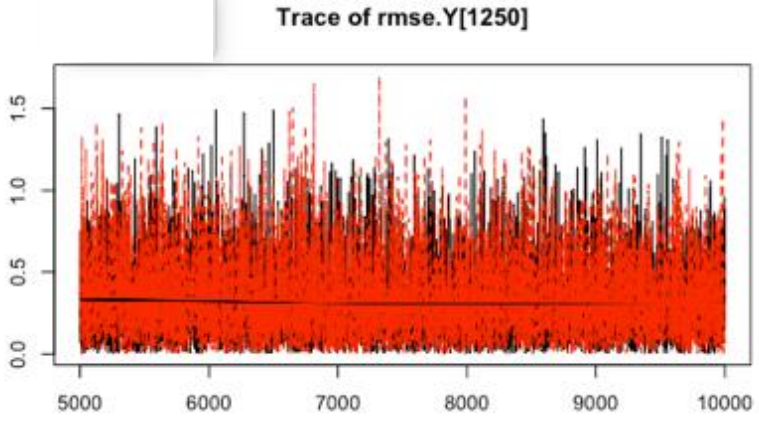
Figure 20: Density plots showing that the 2 chains have not converged for RW2 model: a) $sigma^2_{23}$, and b) $sigma^2_{24}$. This suggests that the RW2 model cause difference variances in the outputs and is not only related to the intervals of missing data.

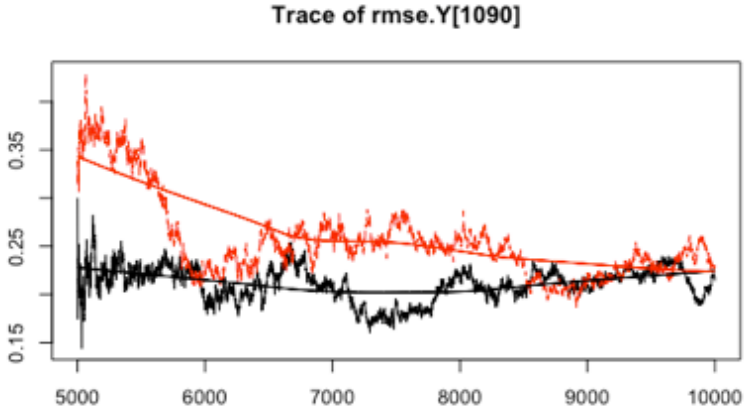


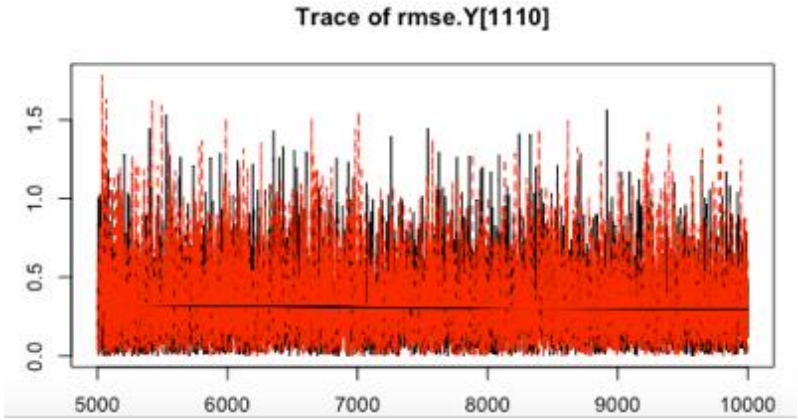

Trace of rmse.Y[1100]
0.0
0.5
1.0
1.5
5000
6000
7000
8000
9000
10000


Trace of rmse.Y[1259]
0.0
0.5
1.0
1.5
5000
6000
7000
8000
9000
10000


Trace of rmse.Y[1099]
0.15
0.25
0.35
5000
6000
7000
8000
9000
10000


Trace of rmse.Y[1250]
0.0
0.5
1.0
1.5
5000
6000
7000
8000
9000
10000


Trace of rmse.Y[1090]
0.15
0.25
0.35
5000
6000
7000
8000
9000
10000


Trace of rmse.Y[1110]
0.0
0.5
1.0
1.5
5000
6000
7000
8000
9000
10000

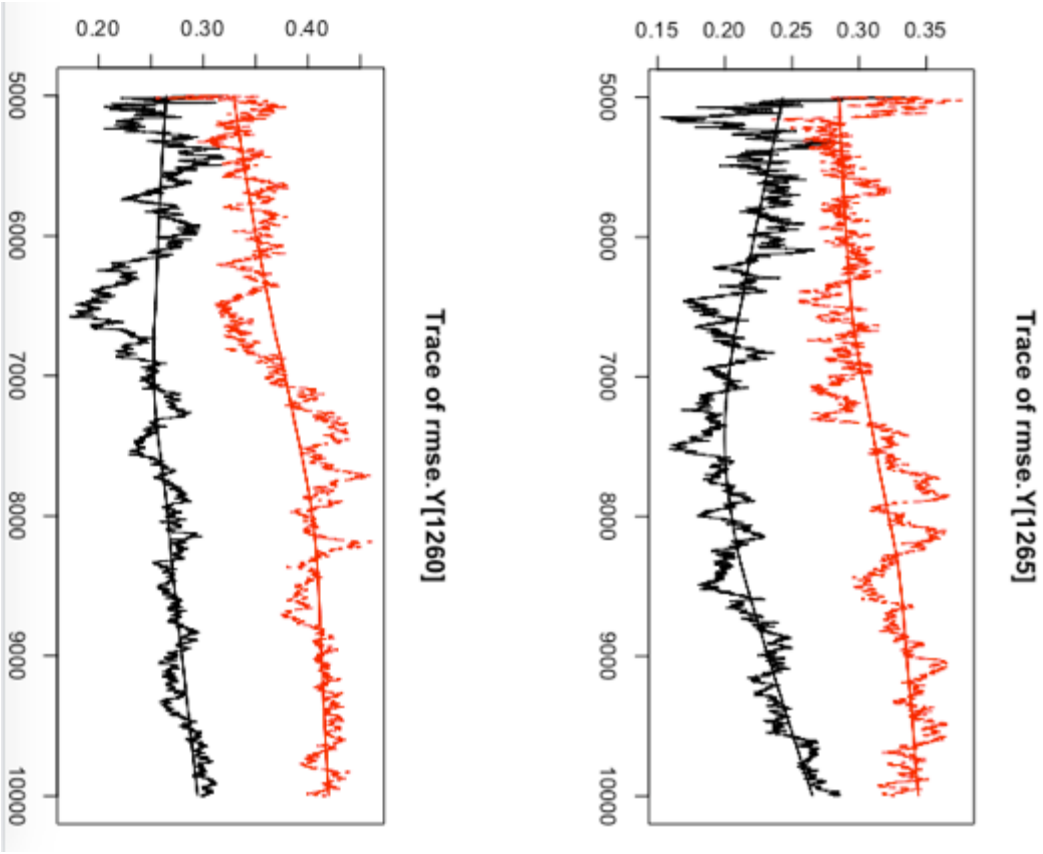


Figure 21: Progression of 11 traceplots showing how the 2 chains do not converge well for deviance, $sigma^2_{23}$ and $sigma^2_{24}$ in an RW1 model. The next 8 show how the RW1 model chains do not converge for RMSE estimate ($rmse.Y_t$) parameters that have associated recorded measurements ($Heathrow_t$), namely rmse.Y1090, rmse.Y1099, rmse.Y1260 and rmse.Y1265, but do converge in the missing data, namely rmse.Y1100, rmse.Y1110, rmse.Y1250 and rmse.Y1259

Consequently, 11 traceplots have been chosen. The first 3 show that amount of converge between the chains for deviance, $sigma^2_1$ and $sigma^2_2$ area all poor in an RW1 model. The next 8 show how the RW1 model chains do not converge for RMSE estimate ($rmse.Y_t$) parameters that have associated recorded measurements ($Heathrow_t$), namely rmse.Y1090, rmse.Y1099, rmse.Y1260 and rmse.Y1265, but converge in the missing data, namely rmse.Y1100, rmse.Y1110, rmse.Y1250 and rmse.Y1259 (Figure 21). This is the opposite way round for the convergence compared to the "true" measurements (Y_t) in the first Heathrow model with RW1 (jag.mod.heathrow) (see Figure 7).

The table of the gelman diagnostics for selected parameters for RW2 model, namely of deviance, $sigma^2_{23}$, $sigma^2_{24}$ and rmse.Y1090 to rmse.Y1110 (Table 7) show that the point estimate and 95% CI estimate are > 1 when there are associated recorded measurements ($Heathrow_t$) and become ~1 when associated with missing data. Further, the ggplot of root mean squared error estimates ($rmse.Y_t$) with associated 95% credible interval boundaries showing that RW2 model has generally high RMSE, compared to RW1 model, with more variable 95% credible interval boundaries for 2000 to 2003 inclusive. Where there are missing data intervals, the random walk model provides a consistent RMSE value of ~0.33 and consistent 95% credible intervals (upper ~0.87; lower ~-0.18) (Figure 22).

	Point est.	Upper C.I.
deviance	1.0120152	1.0198410
sigma2_23	1.0064897	1.0104587
sigma2_24	1.0020322	1.0021891
rmse.Y[1090]	7.6149440	17.3485330
rmse.Y[1091]	6.5732052	14.6020627
rmse.Y[1092]	5.4788395	12.0575798
rmse.Y[1093]	4.4516192	9.7144711
rmse.Y[1094]	3.6633942	7.8246395
rmse.Y[1095]	2.7291312	5.7887964
rmse.Y[1096]	1.9993457	4.0223140
rmse.Y[1097]	1.4204329	2.4265644
rmse.Y[1098]	1.1103618	1.3735207
rmse.Y[1099]	1.0096358	1.0108067
rmse.Y[1100]	1.0001786	1.0012556
rmse.Y[1101]	1.0005288	1.0019129
rmse.Y[1102]	1.0004545	1.0017104
rmse.Y[1103]	0.9999254	0.9999798
rmse.Y[1104]	1.0007827	1.0023367
rmse.Y[1105]	0.9999900	1.0002790
rmse.Y[1106]	0.9999745	1.0001519
rmse.Y[1107]	1.0009745	1.0022686
rmse.Y[1108]	1.0003653	1.0003669
rmse.Y[1109]	1.0001091	1.0008298

Table 7: Results table of the gelman diagnostics for selected parameters for RW2 model, namely of deviance, $sigma^2_{23}$, $sigma^2_{24}$ and rmse.Y1090 to rmse.Y1110 show that the point estimate and 95% CI estimate are > 1 when there are associated recorded measurements ($Heathrow_t$) and become ~1 when associated with missing data

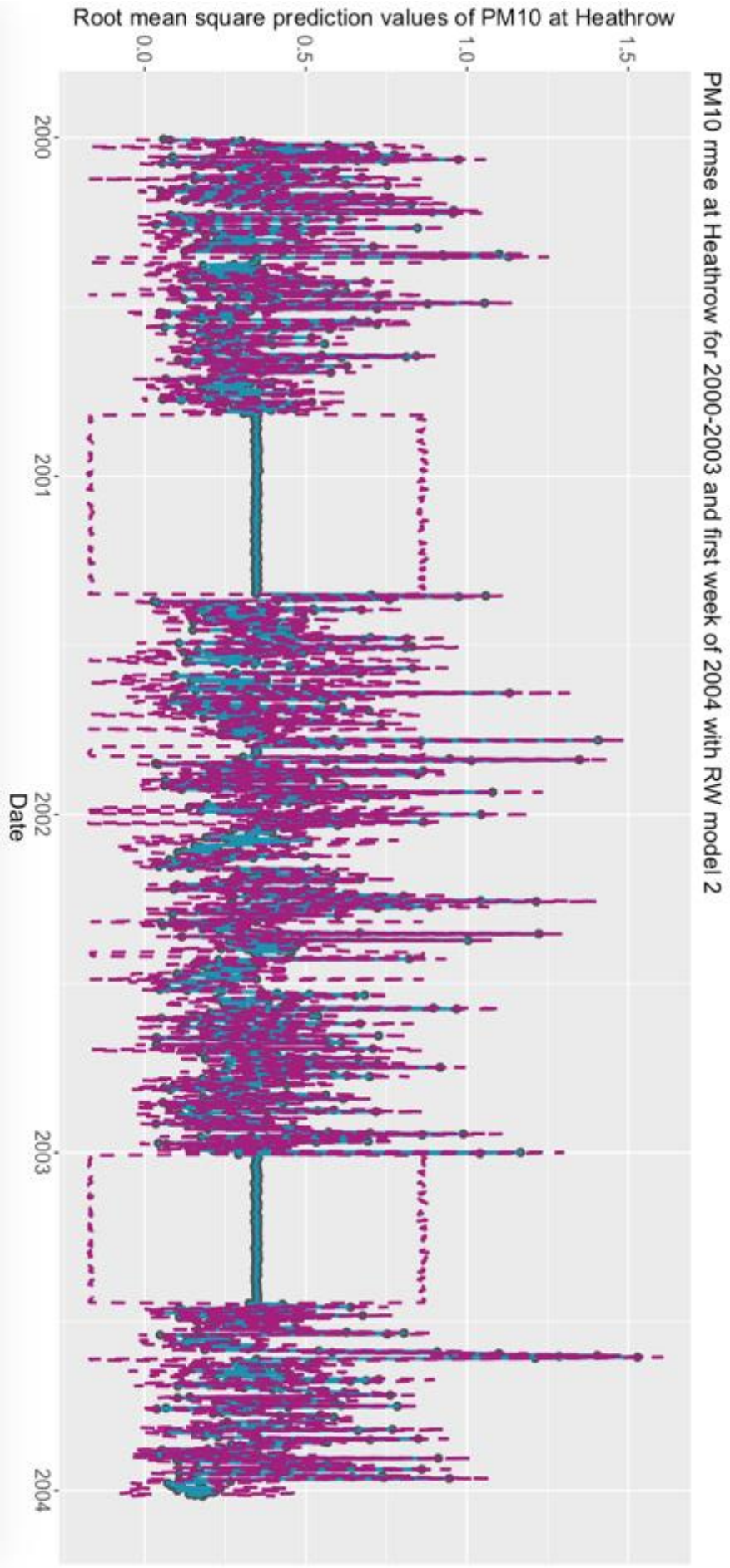


Figure 22: Line and scatter ggplot of root mean squared errors (rmse.Y_t) with associated 95% credible interval boundaries showing that RW2 model has generally high RMSE, compared to RW1 model, with more variable 95% credible interval boundaries for 2000 to 2003 inclusive. Where there are missing data intervals, the random walk model provides a consistent RMSE value of ~0.33 and consistent 95% credible intervals (upper ~0.87; lower ~-0.18).

Results for Random Walk analyses and Bayesian Inference for the Haringey dataset

Random walk analyses and Bayesian inference is undertaken on the Haringey dataset. Initially, the data have been wrangled to remove the 2004 data, so that analysis is only undertaken on data between 2000 and 2003, inclusive, which constitutes 1461 (N3) datapoints (2000 was a leap year). The mean of the Haringey data, without the missing data, is also calculated, so that it can be used to initiate the missing data.

First Haringey model (jags.mod.haringey)

The first Haringey model (jags.mod.haringey) is a random walk of order 1 (RW1) and included as output the parameters: $sigma^2_{13}$, $sigma^2_{14}$, and all Z_t. The Z_t parameters that have associated recorded measurements ($Haringey_t$) show Rhat values of ~ 1, suggesting that the chains would converge, whilst the Z_t parameters that have associated missing data show Rhat values > 1.1, suggesting that they would not converge. The density plots show that there are some minor differences in the estimations of $sigma^2_1$ and $sigma^2_2$ (figures 23a and 23b) for each chain, which may relate to how each chain handled the missing data.

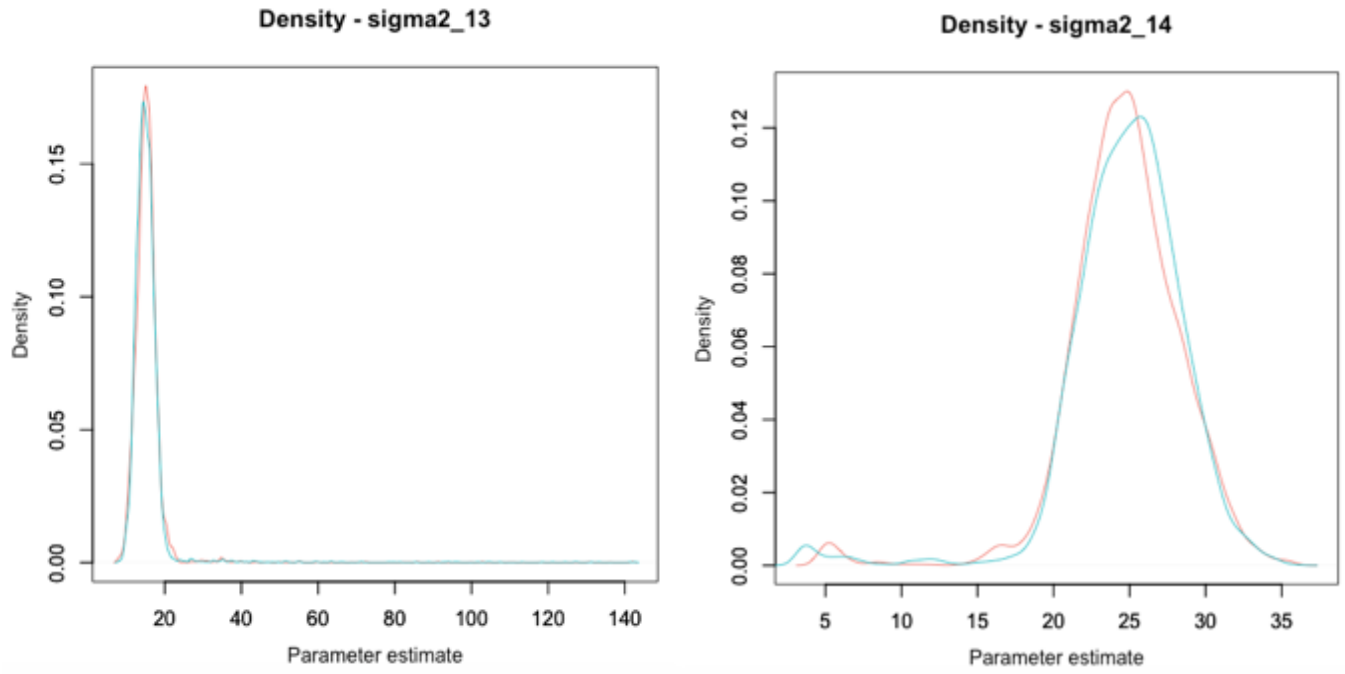
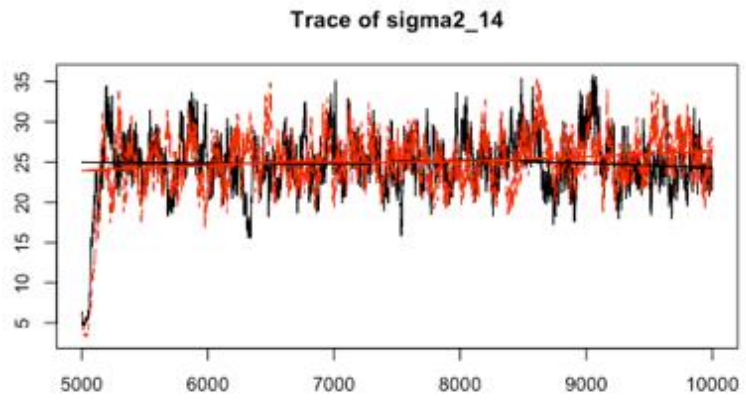
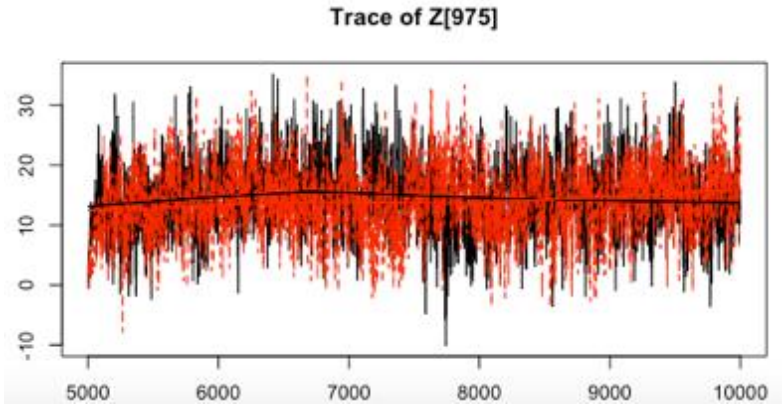


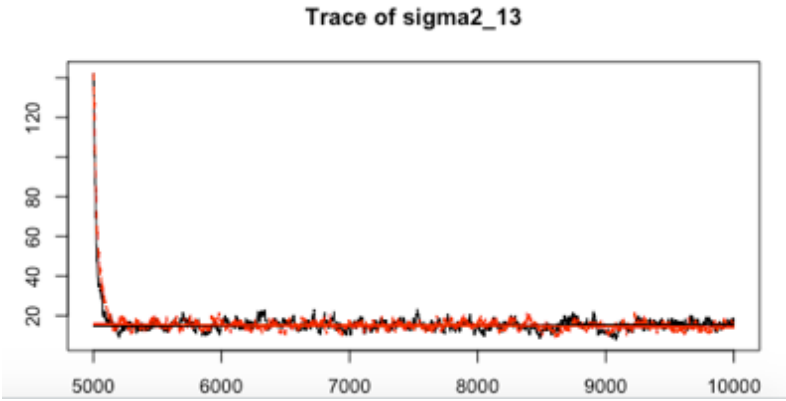
Figure 23: Density plots showing that the 2 chains have not quite converged in RW1 model for: a) $sigma^2_{13}$, and b) $sigma^2_{14}$. This suggests that parts of the data cause difference variances in the outputs, which may result from the intervals of missing data.

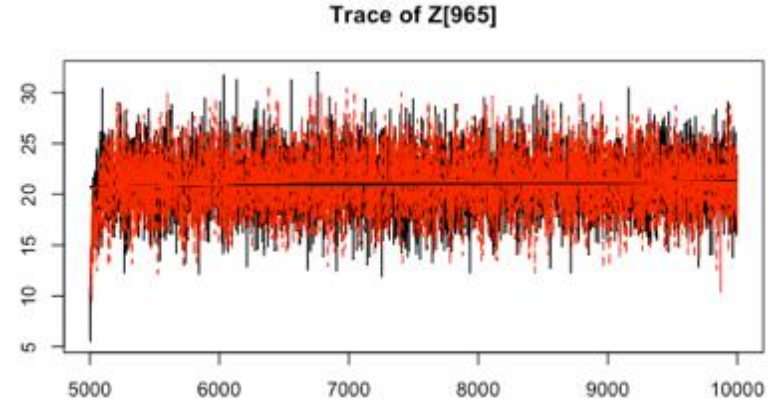
Consequently, 12 traceplots have been chosen. The first 3 show that amount of converge between the chains for deviance, $sigma^2_{13}$ and $sigma^2_{14}$ area all poor in an RW1 model. The next 9 show how the RW1 model chains converge for "true" measurement (Z_t) parameters that have associated recorded measurements ($Haringey_t$), namely Z1 and Z965, lose convergence as the "true" measurements encounter the missing data, namely in Z975 and Z976, stop converging in the missing data, namely Z985 and Z1200, regain convergence as the "true" measurements encounter the recorded measurements again, namely Z1210 and Z1211, and fully converge once again with the recorded measurements, namely Z1219 (Figure 24).

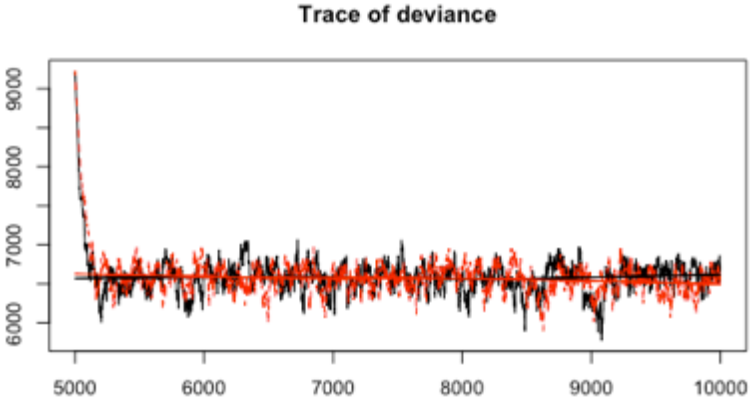
The table of the gelman diagnostics for selected parameters, namely of deviance, $sigma^2_1$, $sigma^2_2$, Z1 and Z1200 to Z1219 (Table 8) show that the point estimate and 95% CI estimate are > 1 whether they are associated either with recorded measurements ($Haringey_t$) or with missing data. Further, the ggplot of the recorded measurements ($Haringey_t$) as well as the "true" measurement parameters (Z_t) with the associated 95% credible interval boundaries shows that the random walk model approximates well to the recorded measurements ($Haringey_t$) with tight 95% credible interval boundaries for 2000 to 2003 inclusive. However, where there are missing data intervals, the random walk model has no constraint and follows a random path until it encounters recorded measurements again, whilst the 95% credible intervals become very wide (Figure 25).

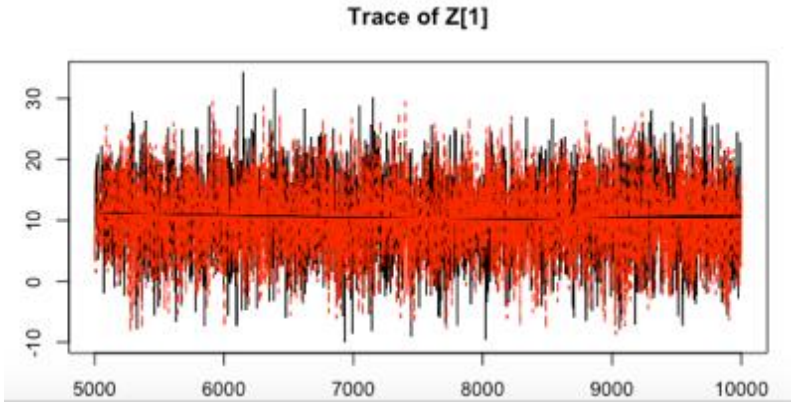

Trace of sigma2_14
5
10
15
20
25
30
35
5000
6000
7000
8000
9000
10000


Trace of Z[975]
-10
0
10
20
30
5000
6000
7000
8000
9000
10000


Trace of sigma2_13
20
40
60
80
120
5000
6000
7000
8000
9000
10000


Trace of Z[965]
5
10
15
20
25
30
5000
6000
7000
8000
9000
10000


Trace of deviance
6000
7000
8000
9000
5000
6000
7000
8000
9000
10000


Trace of Z[1]
-10
0
10
20
30
5000
6000
7000
8000
9000
10000

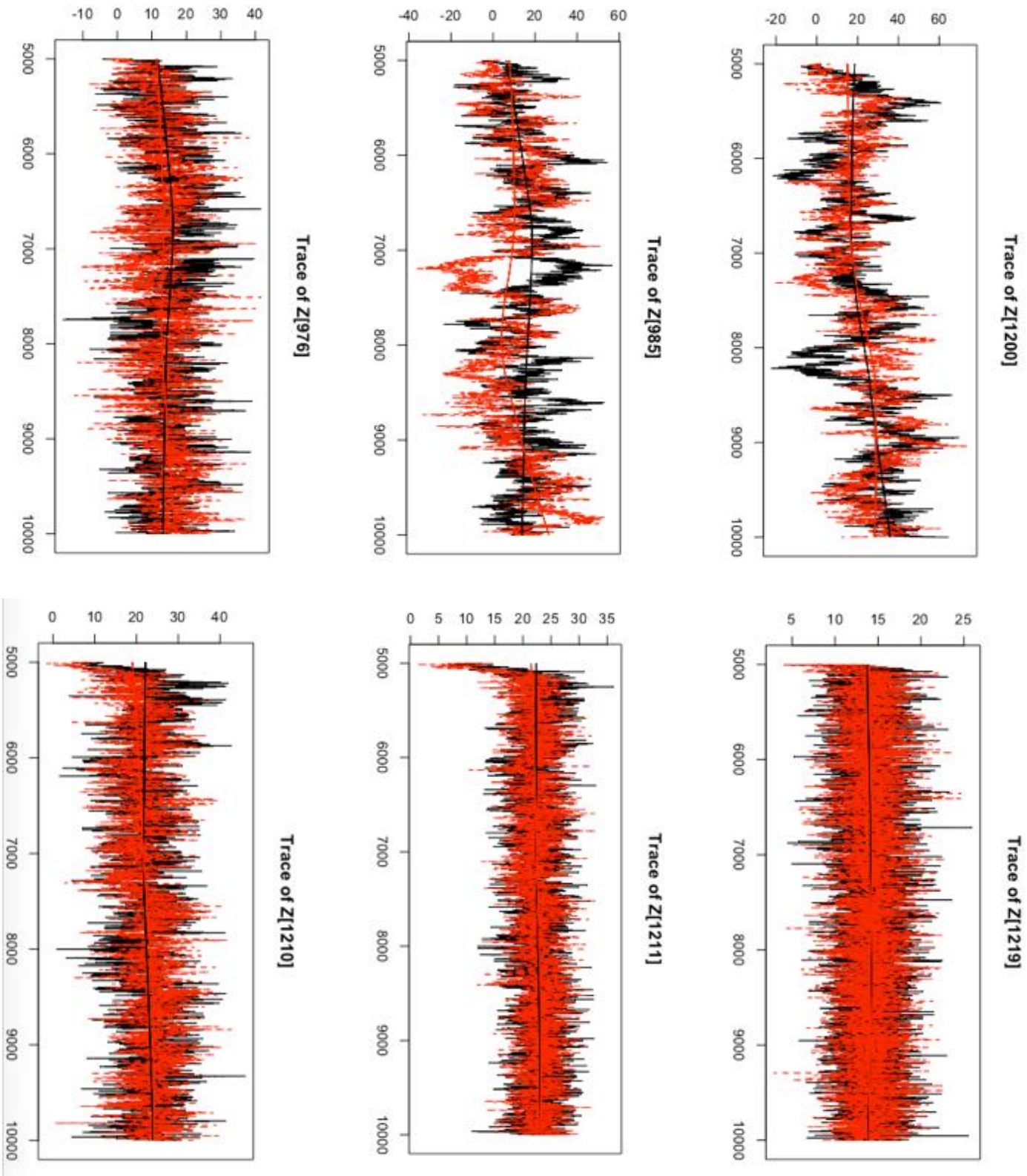


Figure 24: Progression of 12 traceplots showing how the 2 chains do not converge well for deviance, $sigma^2_{13}$ and $sigma^2_{14}$ in an RW1 model. The next 9 show how the RW1 model chains converge for "true" measurement (Z_t) parameters that have associated recorded measurements ($Haringey_t$), namely Z1 and Z965, lose convergence as the "true" measurements encounter the missing data, namely in Z975 and Z976, stop converging in the missing data, namely Z985 and Z1200, regain convergence as the "true" measurements encounter the recorded measurements again, namely Z1210 and Z1211, and fully converge once again with the recorded measurements, namely Z1219

	Point est.	Upper C.I.
deviance	1.0019420	1.0020612
sigma2_13	1.0080763	1.0081868
sigma2_14	1.0011670	1.0026842
Z[1]	1.0001231	1.0009605
Z[1200]	1.0023906	1.0026472
Z[1201]	1.0030194	1.0037235
Z[1202]	1.0027430	1.0040738
Z[1203]	1.0017728	1.0044094
Z[1204]	1.0014679	1.0056888
Z[1205]	1.0018809	1.0075338
Z[1206]	1.0008474	1.0041064
Z[1207]	1.0001746	1.0011102
Z[1208]	0.9999149	0.9999456
Z[1209]	0.9999428	0.9999631
Z[1210]	1.0001189	1.0004331
Z[1211]	1.0013072	1.0022659
Z[1212]	1.0009105	1.0018614
Z[1213]	1.0010464	1.0044262
Z[1214]	1.0006250	1.0035020
Z[1215]	0.9999196	0.9999966
Z[1216]	1.0000580	1.0006511
Z[1217]	1.0000453	1.0004287
Z[1218]	1.0004808	1.0008952
Z[1219]	0.9999630	1.0000974

Table 8: Results table of the gelman diagnostics for selected parameters, namely of deviance, $sigma^2_{13}$, $sigma^2_{14}$, Z1 and Z1200 to Z1219 show that the point estimate and 95% CI estimate are ~1 when there are associated recorded measurements ($Haringey_t$) and become > 1 when associated with missing data

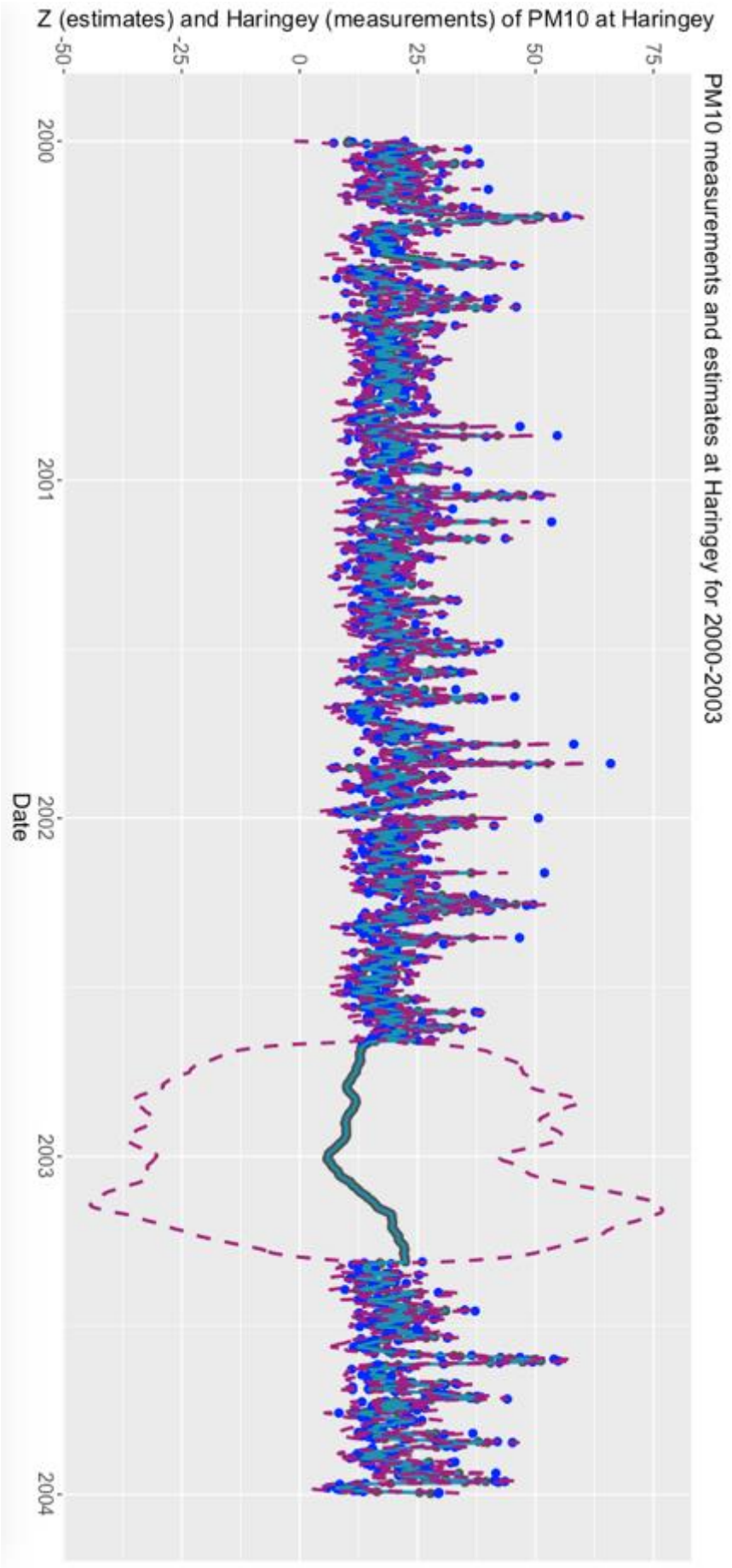


Figure 25: Line and scatter ggplot of recorded measurements ($Haringey_t$) as well as "true" measurement parameters (Z_t) with the associated 95% credible interval boundaries shows that the random walk model approximates well to the recorded measurements ($Haringey_t$) with tight 95% credible interval boundaries for 2000 to 2003 inclusive. However, where there are missing data intervals, the random walk model has no constraint and follows a random path until it encounters recorded measurements again, whilst the 95% credible intervals become very wide.

Second Haringey model (jags.mod.haringey2)

The second Haringey model (jags.mod.haringey2) is a random walk of order 2 (RW2) and included as output the parameters: $sigma^2_1$, $sigma^2_2$, and all Z_t. The Z_t parameters that have associated recorded measurements ($Haringey_t$) and associated missing data both show Rhat values of > 1.1, suggesting that the chains would not converge. The density plots show that there are large differences in the estimations of $sigma^2_{15}$ and $sigma^2_{16}$ (figures 26a and 26b) for each chain, which may relate to how each chain handled the missing data.

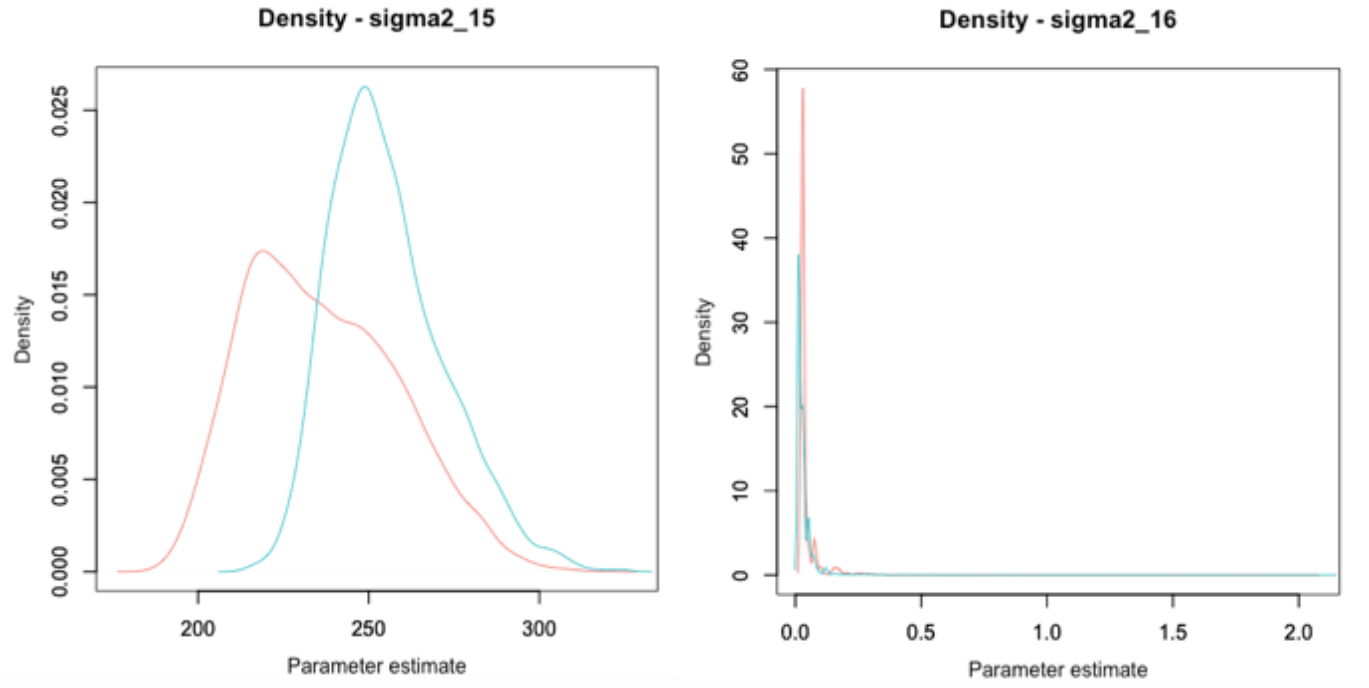
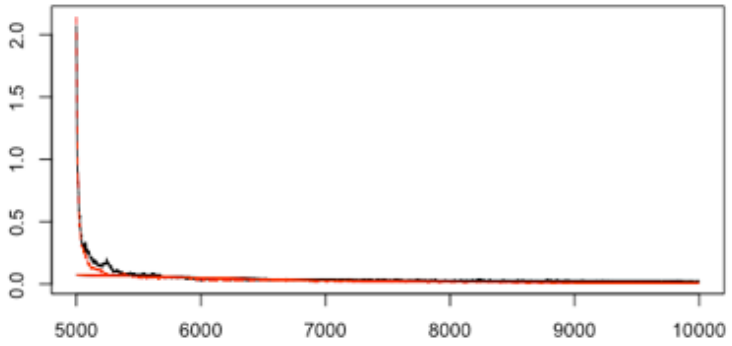


Figure 26: Density plots showing that the 2 chains have not converged for RW2 model: a) $sigma^2_1$, and b) $sigma^2_2$. This suggests that the RW2 model cause difference variances in the outputs and is not only related to the intervals of missing data.

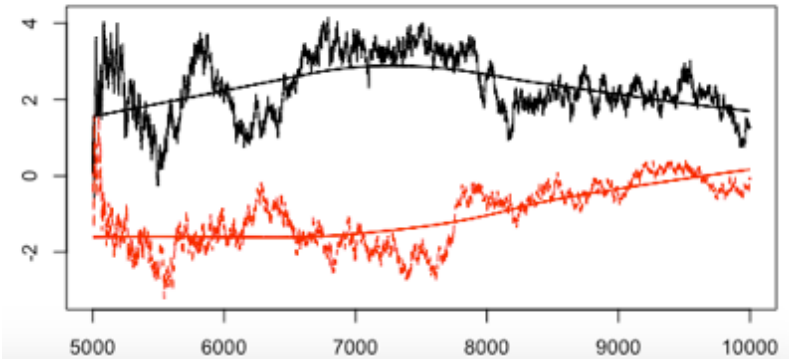
Consequently, 12 traceplots have been chosen. The first 3 show that amount of converge between the chains for deviance, $sigma^2_{15}$ and $sigma^2_{16}$ area all very poor in an RW2 model. The next 9 show how the RW2 model chains do not converge regardless of whether "true" measurement (Z_t) parameters are associated either with recorded measurements ($Haringey_t$), namely in Z1, Z965, Z975, Z1211 and Z1219, or with the missing data, namely in Z976, Z985, Z1200 and Z1210 (Figure 27).

The table of the gelman diagnostics for selected parameters, namely of deviance, $sigma^2_{15}$, $sigma^2_{16}$, Z1 and Z1200 to Z1209 (Table 9) show that the point estimate and 95% CI estimate are ~1 when there are associated recorded measurements ($Haringey_t$) and become > 1 when associated with missing data. Further, the ggplot of the recorded measurements ($Haringey_t$) as well as the "true" measurement parameters (Z_t) with the associated 95% credible interval boundaries shows that the random walk model (RW2) does not approximate to the recorded measurements ($Haringey_t$) and that the 95% credible interval boundaries are variable across the plot. This is shown for 2000 to 2003, inclusive, and suggests that there is no constraint on RW2 model from data or missing data alike (Figure 28). This contrasts with results from the RW1 model for the same period of time (Figure 25).

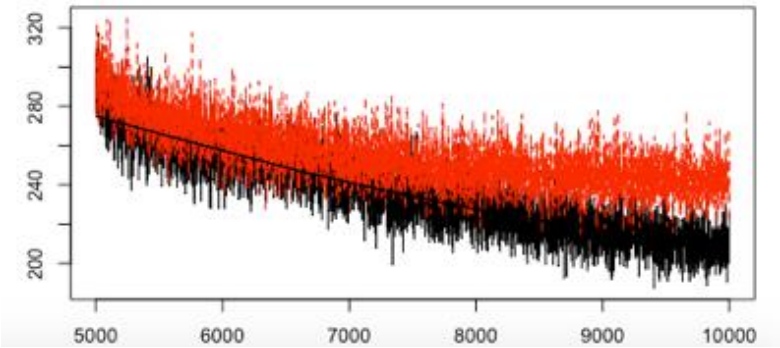
Trace of sigma2_16



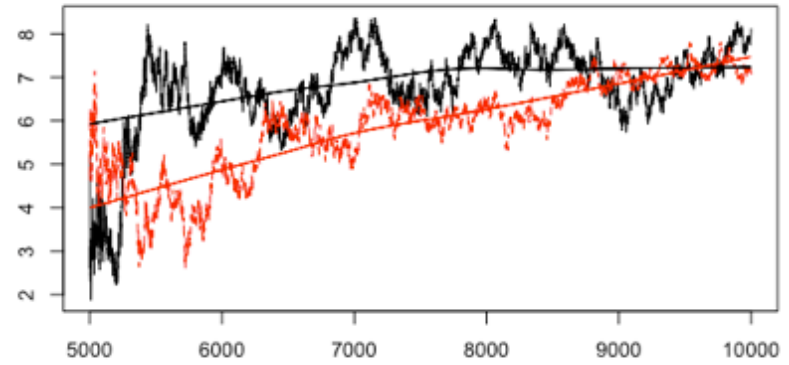
Trace of Z[975]



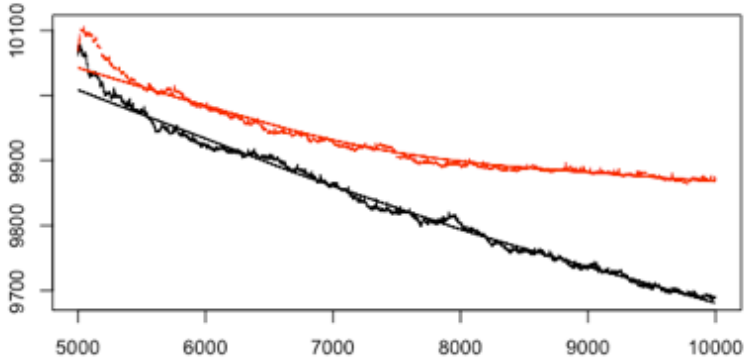
Trace of sigma2_15



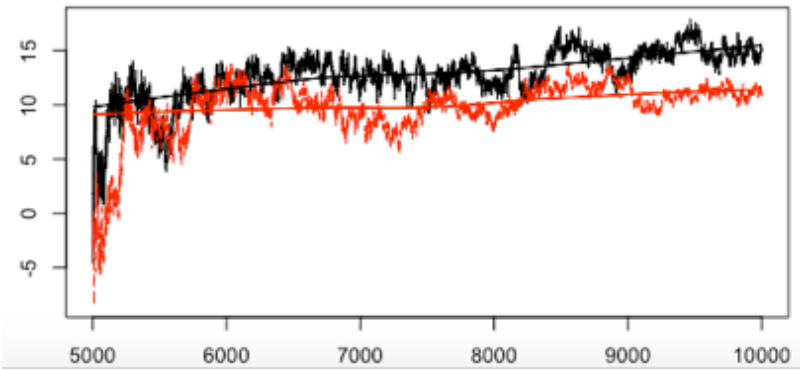
Trace of Z[965]



Trace of deviance



Trace of Z[1]



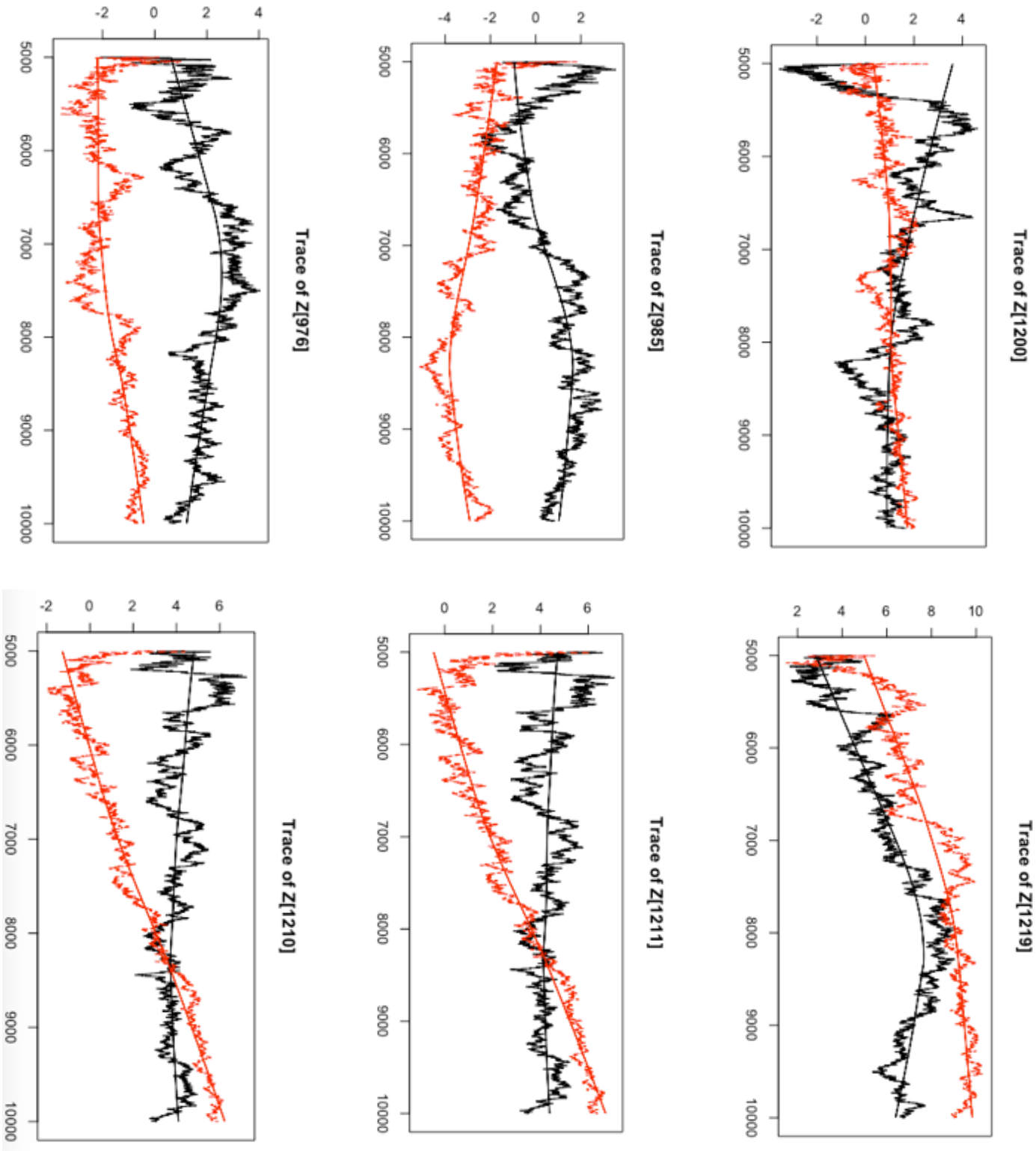


Figure 27: Progression of 12 traceplots showing how the 2 chains do not converge well for deviance, $sigma^2_{15}$ and $sigma^2_{16}$ area in an RW2 model. The next 9 show how the RW2 model chains do not converge regardless of whether "true" measurement (Z_t) parameters are associated either with recorded measurements ($Haringey_t$), namely in Z1, Z965, Z975, Z1211 and Z1219, or with the missing data, namely in Z976, Z985, Z1200 and Z1210.

	Point est.	Upper C.I.
deviance	1.7274833	3.5621511
sigma2_15	1.4931831	2.5495728
sigma2_16	1.0086226	1.0416213
Z[1]	1.6837031	2.9495973
Z[1200]	1.1876721	1.4859350
Z[1201]	1.2620646	1.9578365
Z[1202]	1.4728851	2.4104554
Z[1203]	1.6910030	3.0033956
Z[1204]	1.8442712	3.7013280
Z[1205]	1.9189499	4.3333549
Z[1206]	1.9226316	4.7152499
Z[1207]	1.8823087	4.8648045
Z[1208]	1.8125931	4.8107259
Z[1209]	1.7201120	4.6065788
Z[1210]	1.6038762	4.2549347
Z[1211]	1.4661811	3.6535047
Z[1212]	1.3306565	2.7565001
Z[1213]	1.2306252	1.7305999
Z[1214]	1.1645787	1.1677134
Z[1215]	1.1280345	1.2994334
Z[1216]	1.1676261	1.5932959
Z[1217]	1.3081295	1.9829595
Z[1218]	1.4865776	2.4309377
Z[1219]	1.6574987	2.8642562

Table 9: Results table of the gelman diagnostics for selected parameters, namely of deviance, $sigma^2_{15}$, $sigma^2_{16}$, Z1 and Z1090 to Z1110 show that the point estimate and 95% CI estimate are >1 whether they are associated either with recorded measurements ($Haringey_t$) or with missing data

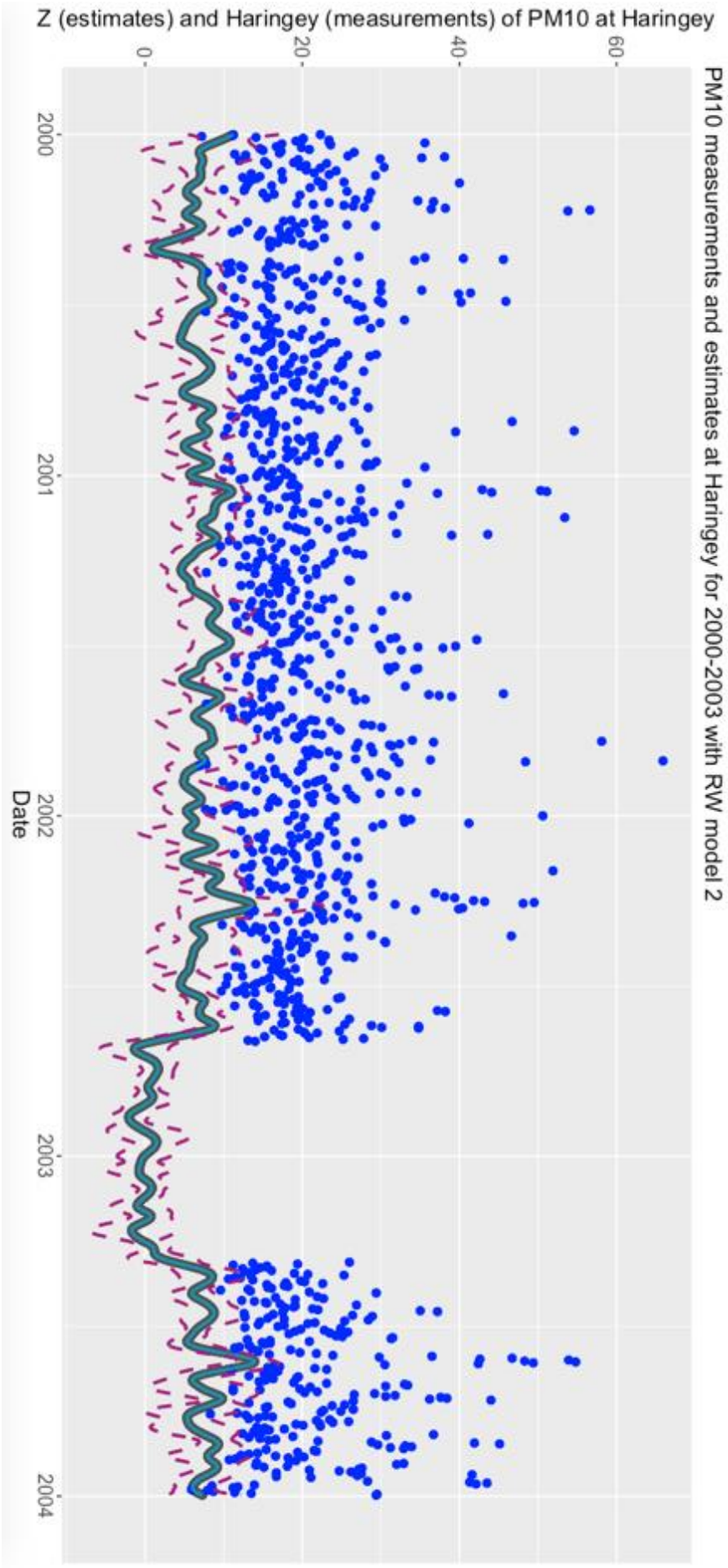


Figure 28: Line and scatter ggplot of the recorded measurements ($Haringey_t$) as well as the "true" measurement parameters (Z_t) with the associated 95% credible interval boundaries shows that the random walk model (RW2) does not approximate to the recorded measurements ($Haringey_t$) and that the 95% credible interval boundaries are variable across the plot. This is shown for 2000 to 2003, inclusive, and suggests that there is no constraint on RW2 model from data or missing data alike

Third Haringey model (jags.mod.haringey3) with informative prior distributions

The third Haringey model (jags.mod.haringey3) is a random walk of order 1 (RW1) and included as output the parameters: $sigma^2_1$, $sigma^2_2$, and all $Z2_t$. The model is built using informative prior distributions for the precision (tau_i) that use the mean and variance of the posterior $sigma^2_i$ estimates from the first Haringey model. These estimates have been transformed into precision estimates and applied to the precision estimates (tau_i) for this third (RW1) Haringey model using a normal distribution, and are given by:

$$
\begin{aligned}
tau17 \sim dnorm(0.064, 0.020) \quad & \# \; normal\ measurement\ error\ precision\ model \\
sigma2_{17} = 1/tau17 \quad & \# \; measurement\ error\ variance \\
tau18 \sim dnorm(0.040, 0.064) \quad & \# \; normal\ estimate\ error\ precision\ model \\
sigma2_{18} = 1/tau18 \quad & \# \; estimate\ error\ variance
\end{aligned}
$$

With the model run using the informative priors, the $Z2_t$ parameters that have associated recorded measurements ($Haringey_t$) show Rhat values of ~ 1, suggesting that the chains would converge, whilst the Z_t parameters that have associated missing data show Rhat values >1.1, suggesting that they would not converge. The density plots show that there are some minor differences in the estimations of $sigma^2_{17}$ and $sigma^2_{18}$ (figures 29a and 29b) for each chain, which may relate to how each chain handled the missing data.

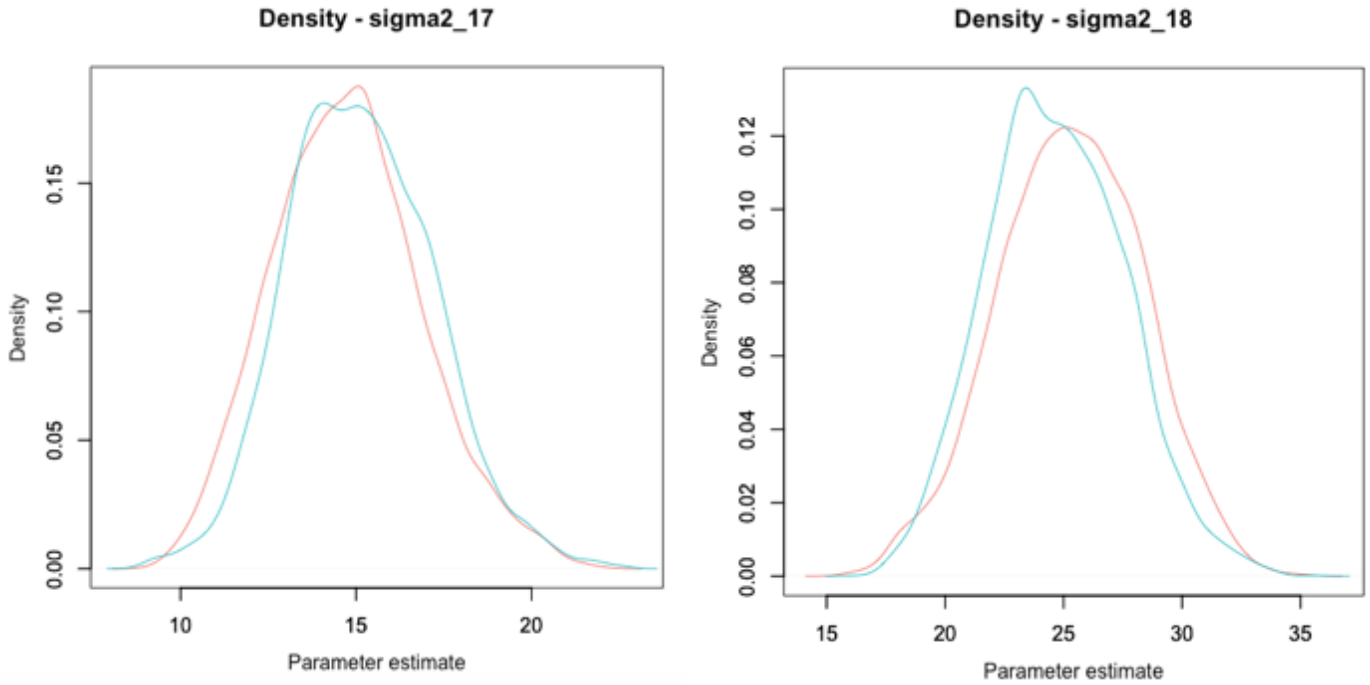
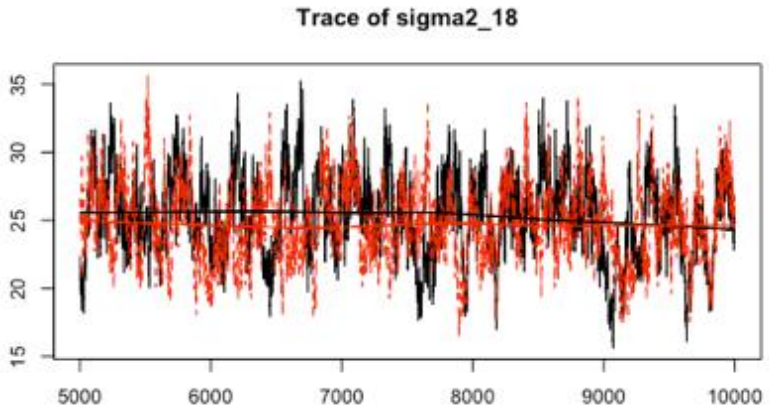
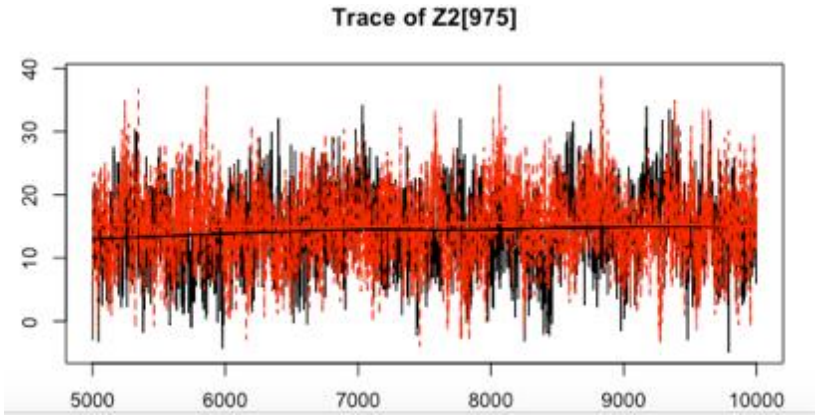


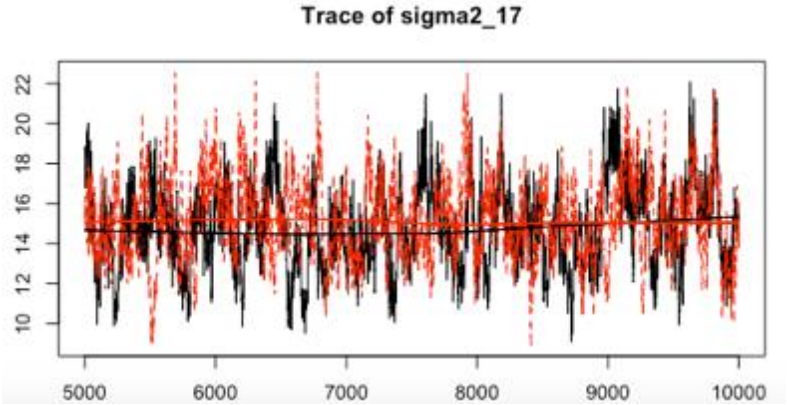
Figure 29: Density plots showing that the 2 chains have not quite converged in RW1 model for: a) $sigma^2_{17}$, and b) $sigma^2_{18}$. This suggests that parts of the data cause difference variances in the outputs, which may result from the intervals of missing data.

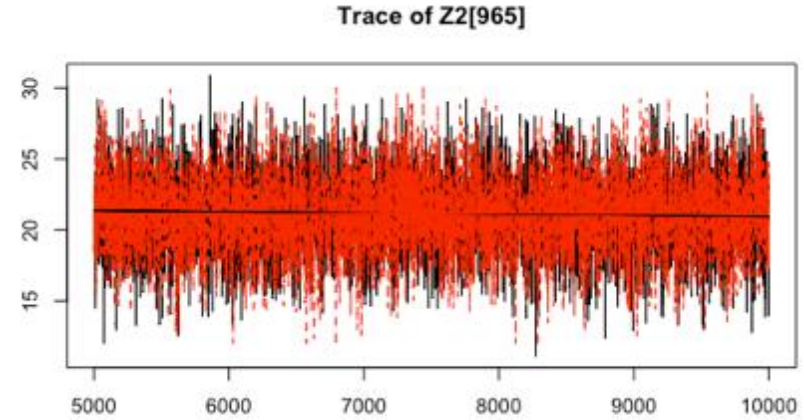
Consequently, 12 traceplots have been chosen. The first 3 show that amount of converge between the chains for deviance, $sigma^2_{17}$ and $sigma^2_{18}$ area all poor in an RW1 model. The next 9 traceplots show that the RW1 model chains converge for "true" measurement (Z_t) parameters that have associated recorded measurements ($Haringey_t$), namely Z2 1 and Z2 965, lose convergence as the "true" measurements encounter the missing data, namely in Z2 975 and Z2 976, stop converging in the missing data, namely Z2 985 and Z2 1200, regain convergence as the "true" measurements encounter the recorded measurements again, namely Z2 1210 and Z2 1211, and fully converge once again with the recorded measurements, namely Z2 1219 (Figure 30).

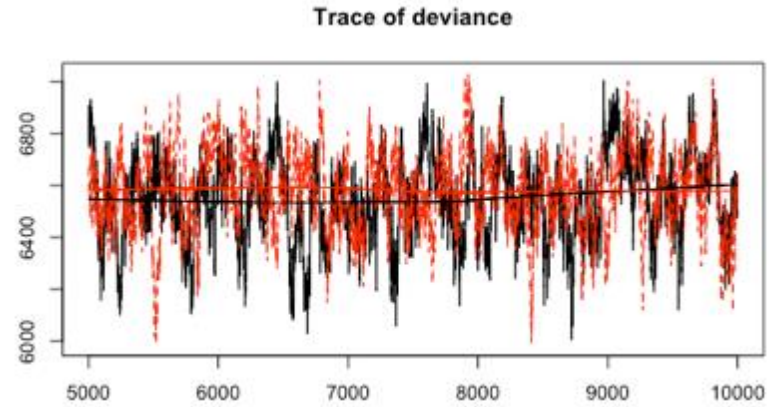
The table of the gelman diagnostics for selected parameters, namely of deviance, $sigma^2_{17}$, $sigma^2_{18}$, Z2 1 and Z2 1200 to Z2 1219 (Table 10) show that the point estimate and 95% CI estimate are > 1 whether they are associated either with recorded measurements ($Haringey_t$) or with missing data. Further, the ggplot of the recorded measurements ($Haringey_t$) as well as the "true" measurement parameters ($Z2_t$) with the associated 95% credible interval boundaries shows that the random walk model approximates well to the recorded measurements ($Haringey_t$) with tight 95% credible interval boundaries for 2000 to 2003 inclusive. However, where there are missing data intervals, the random walk model has no constraint and follows a random path until it encounters recorded measurements again, whilst the 95% credible intervals become very wide (Figure 31).

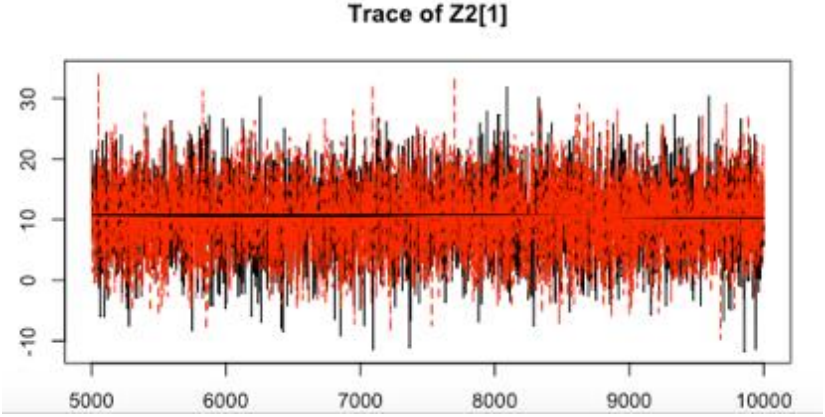

Trace of sigma2_18
35
30
25
20
15
5000
6000
7000
8000
9000
10000


Trace of Z2[975]
40
30
20
10
0
5000
6000
7000
8000
9000
10000


Trace of sigma2_17
22
20
18
16
14
12
10
5000
6000
7000
8000
9000
10000


Trace of Z2[965]
30
25
20
15
5000
6000
7000
8000
9000
10000


Trace of deviance
6800
6400
6000
5000
6000
7000
8000
9000
10000


Trace of Z2[1]
30
20
10
0
-10
5000
6000
7000
8000
9000
10000

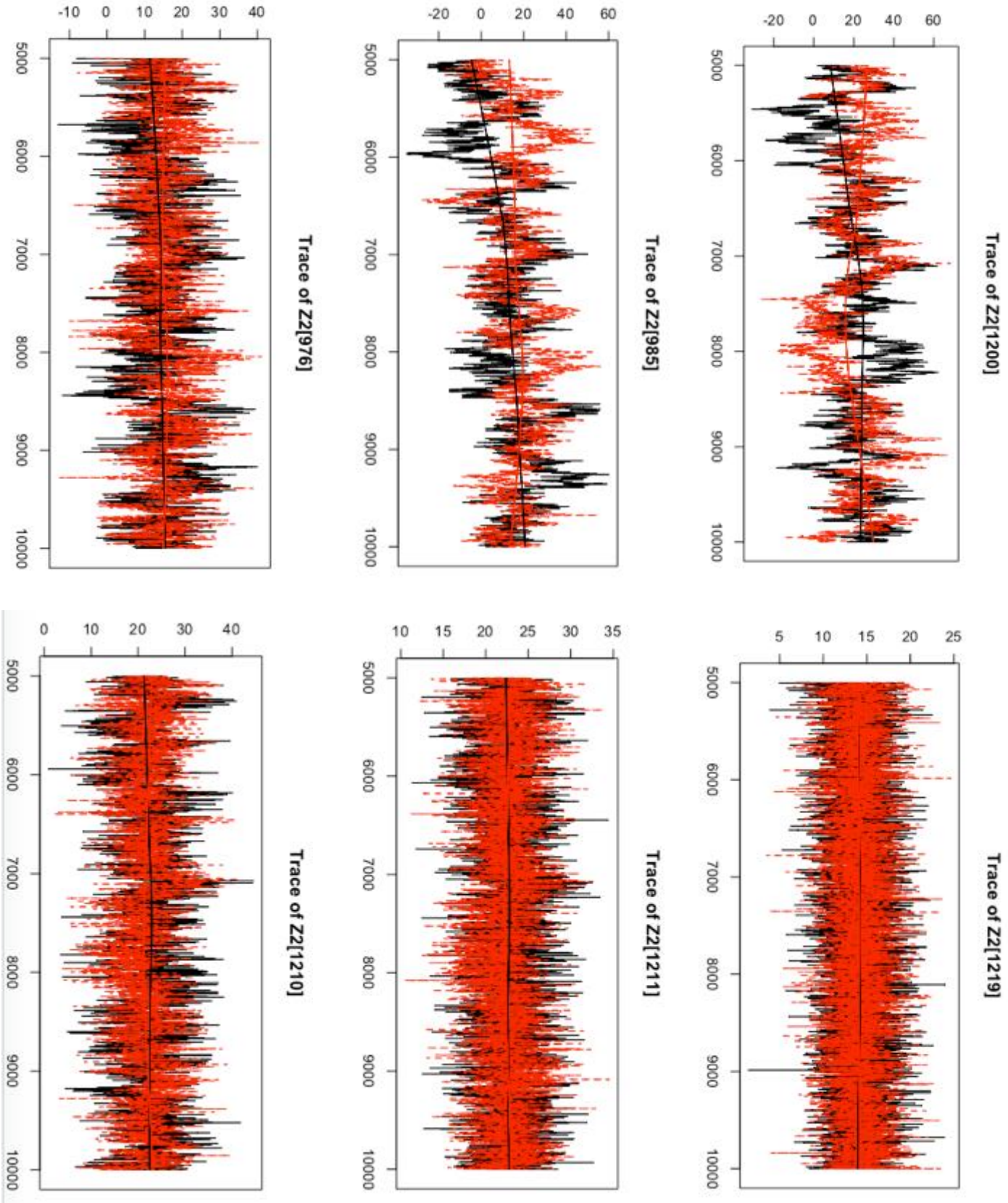


Figure 30: Progression of 12 traceplots showing how the 2 chains do not converge well for deviance, $sigma^2_{17}$ and $sigma^2_{18}$ in an RW1 model. The next 9 traceplots show that the RW1 model chains converge for "true" measurement ($Z2_t$) parameters that have associated recorded measurements ($Haringey_t$), namely Z2 1 and Z2 965, lose convergence as the "true" measurements encounter the missing data, namely in Z2 975 and Z2 976, stop converging in the missing data, namely Z2 985 and Z2 1200, regain convergence as the "true" measurements encounter the recorded measurements again, namely Z2 1210 and Z2 1211, and fully converge once again with the recorded measurements, namely Z2 1219

	Point est.	Upper C.I.
deviance	1.0062587	1.0062827
sigma2_17	1.0035502	1.0036912
sigma2_18	1.0081502	1.0085462
Z2[1]	1.0001910	1.0012808
Z2[1200]	1.0382839	1.1468328
Z2[1201]	1.0414977	1.1596083
Z2[1202]	1.0464895	1.1794044
Z2[1203]	1.0524708	1.2062226
Z2[1204]	1.0571011	1.2237749
Z2[1205]	1.0634583	1.2500792
Z2[1206]	1.0641316	1.2515917
Z2[1207]	1.0549245	1.2186214
Z2[1208]	1.0472542	1.1912870
Z2[1209]	1.0353639	1.1442473
Z2[1210]	1.0186102	1.0817790
Z2[1211]	1.0045307	1.0223585
Z2[1212]	1.0001770	1.0008275
Z2[1213]	1.0004384	1.0021018
Z2[1214]	0.9999874	1.0003249
Z2[1215]	0.9999499	1.0000856
Z2[1216]	0.9999520	1.0000466
Z2[1217]	1.0002274	1.0007615
Z2[1218]	1.0004782	1.0021659
Z2[1219]	1.0005345	1.0028975

Table 10: Results table of the gelman diagnostics for selected parameters, namely of deviance, $sigma^2_{17}$, $sigma^2_{18}$, Z2 1 and Z2 1200 to Z2 1219 show that the point estimate and 95% CI estimate are ~1 when there are associated recorded measurements ($Haringey_t$) and become > 1 when associated with missing data

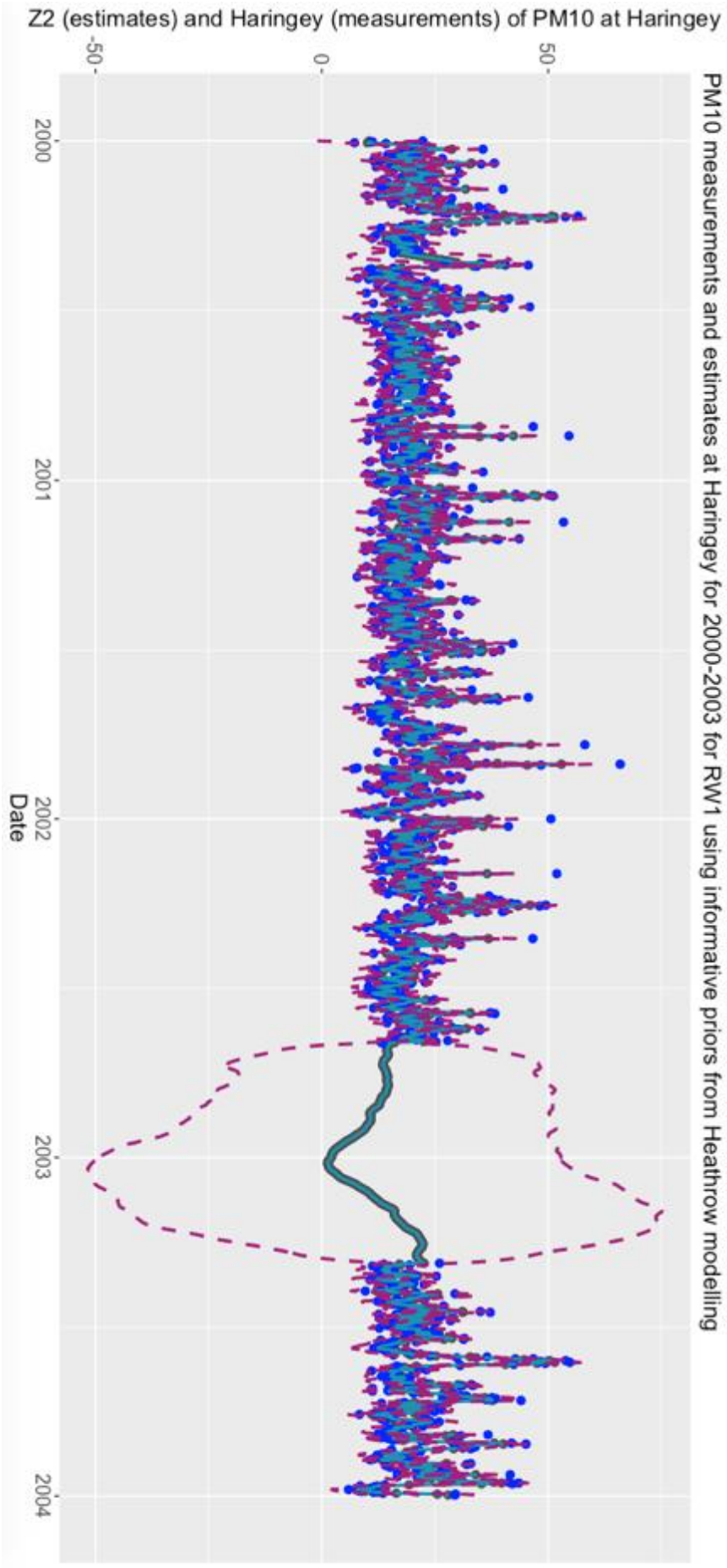


Figure 31: Line and scatter ggplot of recorded measurements ($Haringey_t$) as well as "true" measurement parameters ($Z2_t$) with the associated 95% credible interval boundaries shows that the random walk model approximates well to the recorded measurements ($Haringey_t$) with tight 95% credible interval boundaries for 2000 to 2003 inclusive. However, where there are missing data intervals, the random walk model has no constraint and follows a random path until it encounters recorded measurements again, whilst the 95% credible intervals become very wide.

Fourth Haringey model (jags.mod.haringey4) with informative prior distributions

The fourth Haringey model (jags.mod.haringey4) is a random walk of order 2 (RW2) and included as output the parameters: $sigma^2_1$, $sigma^2_2$ and all $Z2_t$. The model is built using informative prior distributions for the precision (tau_i) that use the mean and variance of the posterior $sigma^2_i$ estimates from the second Haringey model. These estimates have been transformed into precision estimates and applied to the precision estimates (tau_i) for this fourth (RW2) Haringey model using a normal distribution, and are given by:

$$
\begin{aligned}
tau19 \sim dnorm(0.004, 0.002) \quad & \# \; normal\ measurement\ error\ precision\ model \\
sigma2_{19} = 1/tau19 \quad & \# \; measurement\ error\ variance \\
tau20 \sim dnorm(0.002, 156.25) \quad & \# \; normal\ estimate\ error\ precision\ model \\
sigma2_{20} = 1/tau20 \quad & \# \; estimate\ error\ variance
\end{aligned}
$$

With the model run using the informative priors, the $Z2_t$ parameters that have associated recorded measurements ($Haringey_t$) show Rhat values of ~ 1, suggesting that the chains are likely to converge, whilst the $Z2_t$ parameters that have associated missing data show Rhat values > 1.1, suggesting that they would not converge. This convergence situation is much better than for the second Haringey model (jags.mod.haringey2) is with a RW2 model. The density plots show that there are small differences in the estimations of $sigma^2_{19}$ and $sigma^2_{20}$ (figures 32a and 32b) for each chain, which may relate to how each chain handled the missing data.

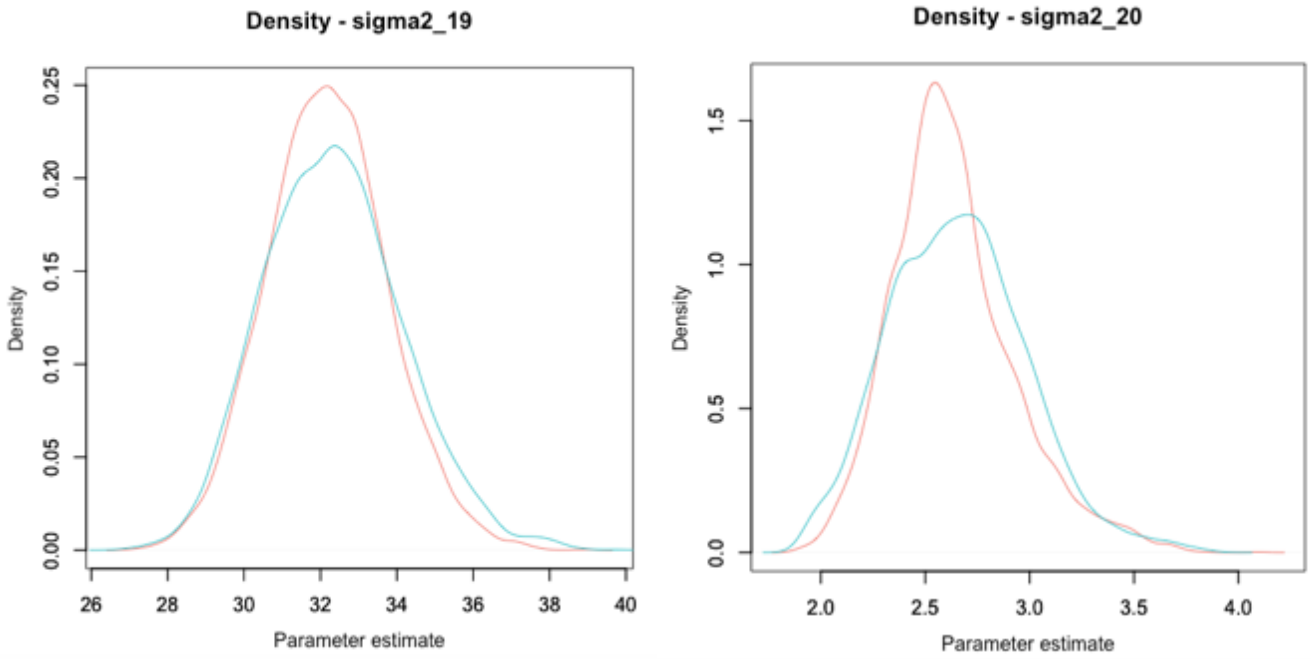
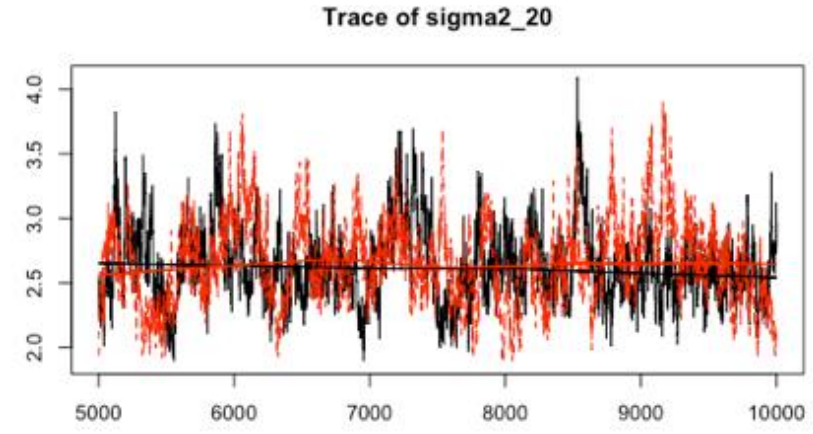
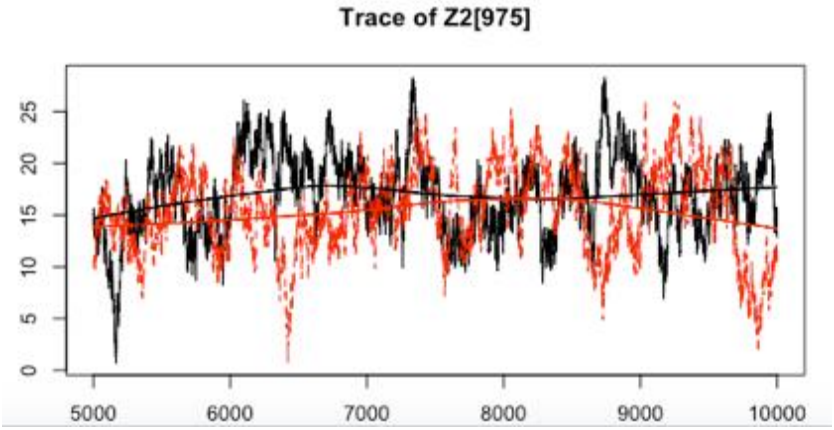


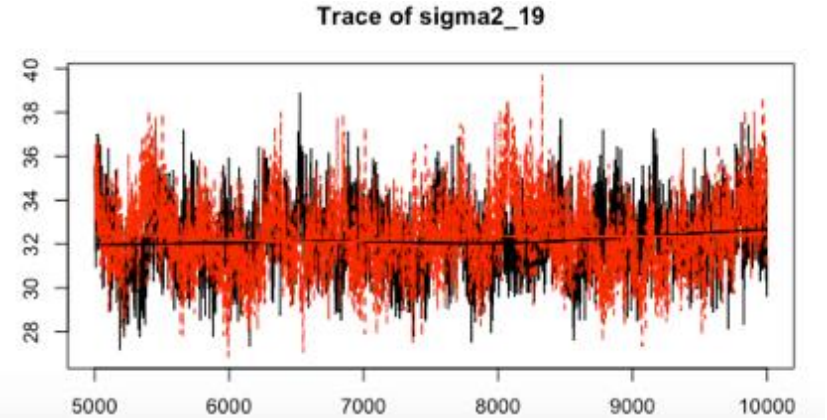
Figure 32: Density plots showing that the 2 chains have almost converged for the RW2 model: a) $sigma^2_{19}$, and b) $sigma^2_{20}$. This suggests that the RW2 model cause difference variances in the outputs and is not only related to the intervals of missing data.

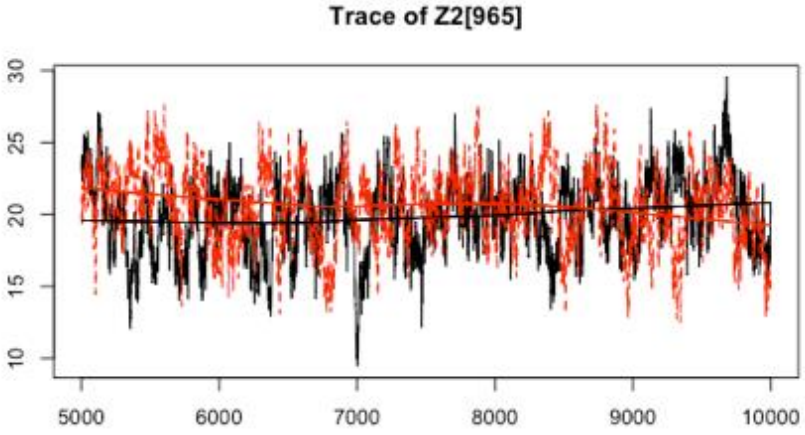
Consequently, 12 traceplots have been chosen. The first 3 show that amount of converge between the chains for deviance, $sigma^2_1$ and $sigma^2_2$ are a little poor for this RW2 model with informative priors. The next 9 traceplots show that the RW2 model chains almost converge for "true" measurement ($Z2_t$) parameters that have associated recorded measurements ($Haringey_t$), namely Z2 1 and Z2 965, lose convergence as the "true" measurements encounter the missing data, namely in Z2 975 and Z2 976, stop converging in the missing data, namely Z2 985 and Z2 1200, regain some convergence as the "true" measurements encounter recorded measurements again, namely Z2 1210 and Z2 1211, and almost converge once again with the recorded measurements, namely Z2 1219 (Figure 33).

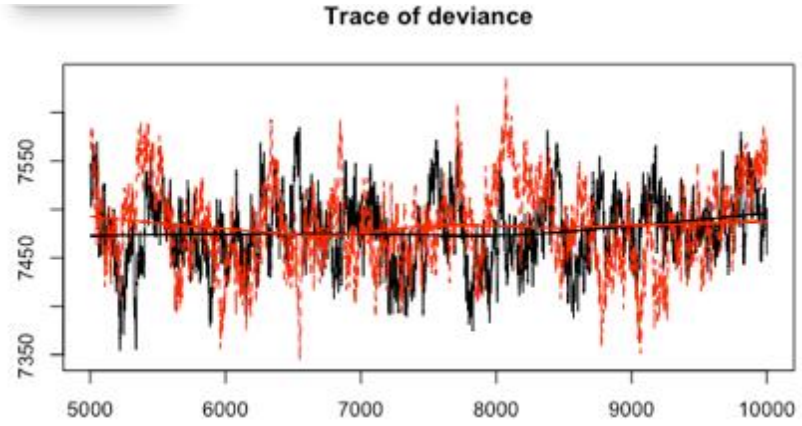
The table of the gelman diagnostics for selected parameters, namely of deviance, $sigma^2_{19}$, $sigma^2_{20}$, Z2 1 and Z2 1200 to Z2 1219 (Table 11) show that the point estimate and 95% CI estimate are ~1 when there are associated recorded measurements ($Haringey_t$) and become > 1 when associated with missing data. Further, the ggplot of recorded measurements ($Haringey_t$) as well as "true" measurement parameters ($Z2_t$) with the associated 95% credible interval boundaries shows that the random walk model approximates well to the recorded measurements ($Haringey_t$) with tight 95% credible interval boundaries for 2000 to 2003 inclusive. However, where there are missing data intervals, the random walk model has no constraint and follows a random path until it encounters recorded measurements again, whilst the 95% credible intervals become wide (Figure 34).

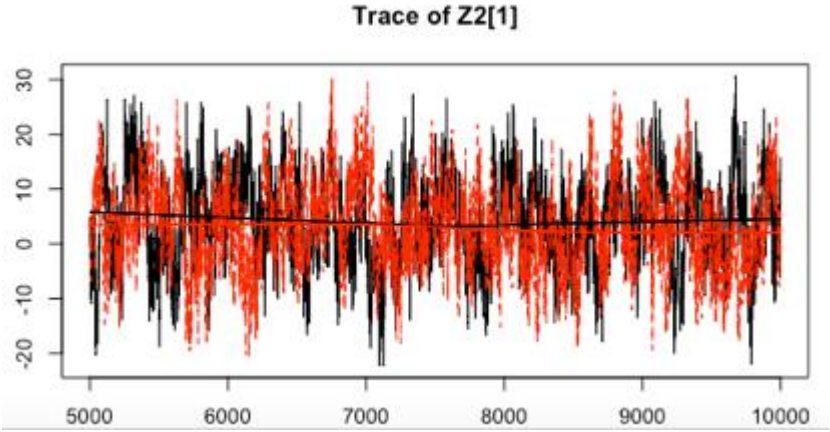

Trace of sigma2_20
4.0
3.5
3.0
2.5
2.0
5000
6000
7000
8000
9000
10000


Trace of Z2[975]
25
20
15
10
5
0
5000
6000
7000
8000
9000
10000


Trace of sigma2_19
40
38
36
34
32
30
28
5000
6000
7000
8000
9000
10000


Trace of Z2[965]
30
25
20
15
10
5000
6000
7000
8000
9000
10000


Trace of deviance
7550
7450
7350
5000
6000
7000
8000
9000
10000


Trace of Z2[1]
30
20
10
0
-10
-20
5000
6000
7000
8000
9000
10000

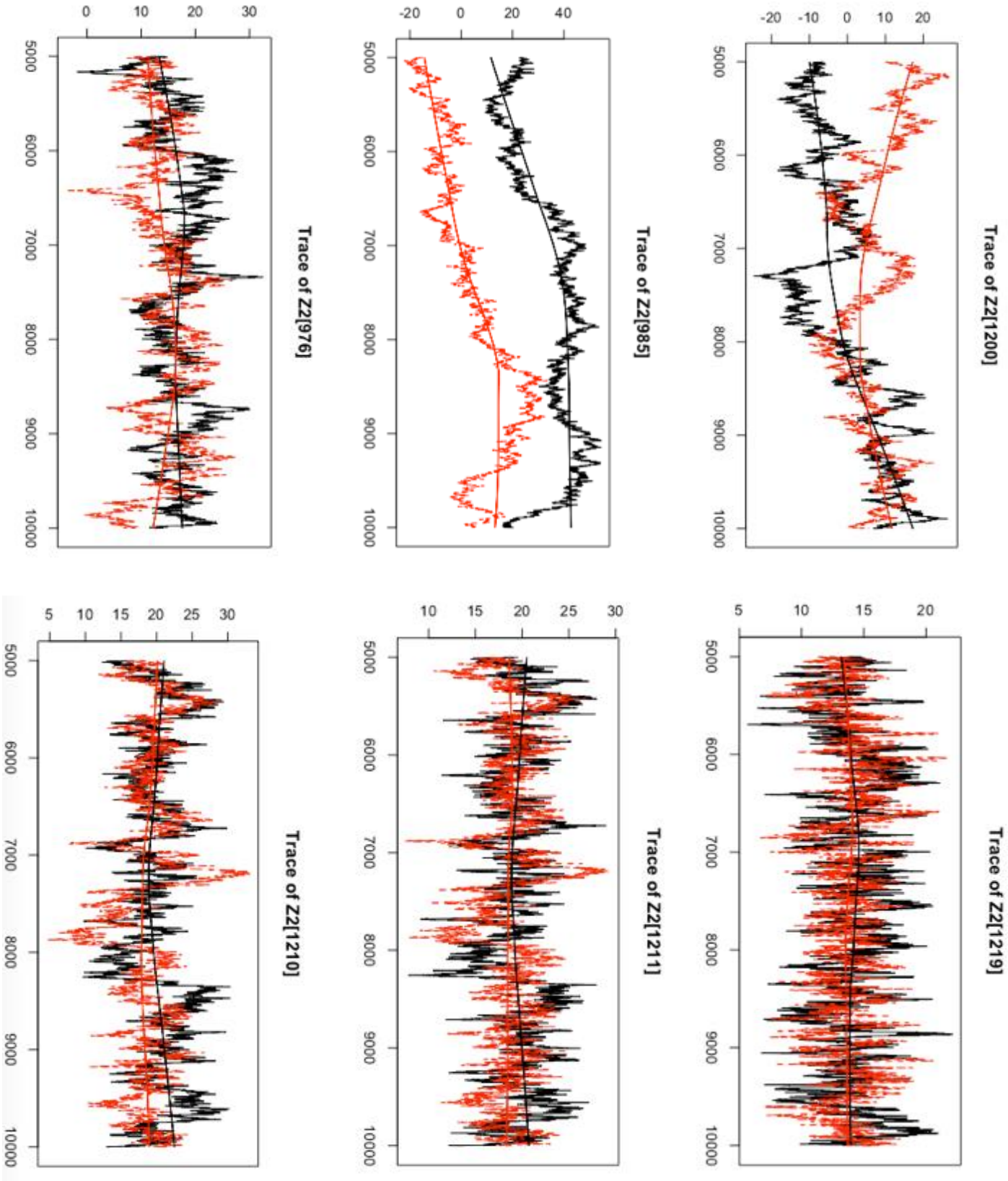


Figure 33: Progression of 12 traceplots showing how the 2 chains do not converge well for deviance, $sigma^2_{19}$ and $sigma^2_{20}$ in an RW2 model. The next 9 traceplots show that the RW2 model chains almost converge for "true" measurement ($Z2_t$) parameters that have associated recorded measurements ($Haringey_t$), namely Z2 1 and Z2 965, lose convergence as the "true" measurements encounter the missing data, namely in Z2 975 and Z2 976, stop converging in the missing data, namely Z2 985 and Z2 1200, regain some convergence as the "true" measurements encounter the recorded measurements again, namely Z2 1210 and Z2 1211, and almost converge once again with the recorded measurements, namely Z2 1219

	Point est.	Upper C.I.
deviance	1.0230682	1.0383558
sigma2_19	1.0120377	1.0246432
sigma2_20	1.0079865	1.0160218
Z2[1]	1.0060384	1.0294249
Z2[1200]	1.3883103	2.2711224
Z2[1201]	1.1490055	1.5279935
Z2[1202]	1.0464941	1.1263374
Z2[1203]	1.0190775	1.0194786
Z2[1204]	1.0253276	1.0838976
Z2[1205]	1.0464758	1.1875655
Z2[1206]	1.0633250	1.2545961
Z2[1207]	1.0681601	1.2734048
Z2[1208]	1.0686140	1.2741197
Z2[1209]	1.0654850	1.2609602
Z2[1210]	1.0572576	1.2327129
Z2[1211]	1.0368421	1.1600857
Z2[1212]	1.0126723	1.0603426
Z2[1213]	1.0032938	1.0134936
Z2[1214]	1.0030249	1.0078258
Z2[1215]	1.0025220	1.0109577
Z2[1216]	1.0038172	1.0167583
Z2[1217]	1.0090360	1.0207996
Z2[1218]	1.0139649	1.0302808
Z2[1219]	1.0147138	1.0401826

Table 11: Results table of the gelman diagnostics for selected parameters, namely of deviance, $sigma^2_{19}$, $sigma^2_{20}$, Z2 1 and Z2 1200 to Z2 1219 show that the point estimate and 95% CI estimate are ~1 when there are associated recorded measurements ($Haringey_t$) and become > 1 when associated with missing data

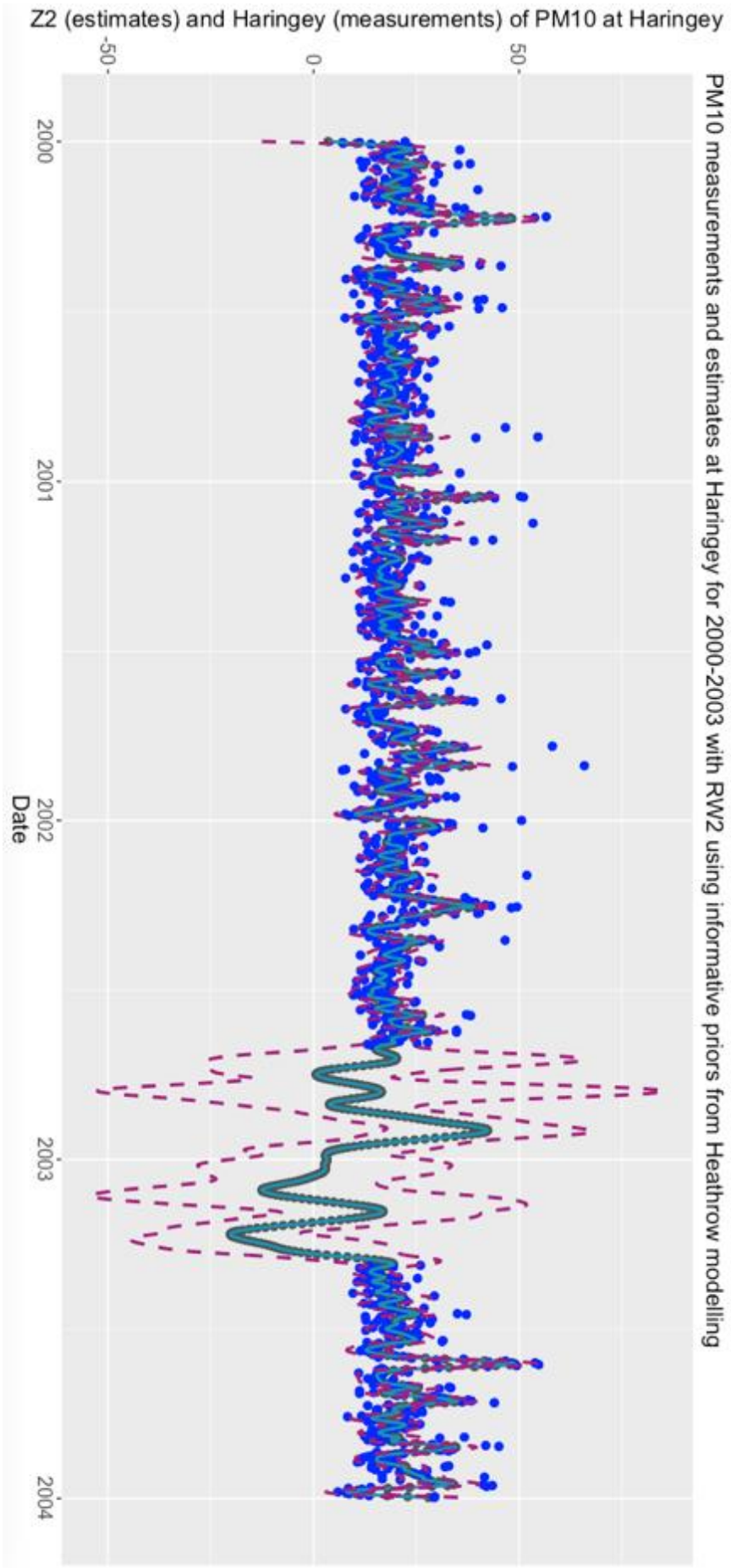


Figure 34: Line and scatter ggplot of recorded measurements ($Haringey_t$) as well as "true" measurement parameters ($Z2_t$) with the associated 95% credible interval boundaries shows that the random walk model approximates well to the recorded measurements ($Haringey_t$) with tight 95% credible interval boundaries for 2000 to 2003 inclusive. However, where there are missing data intervals, the random walk model has no constraint and follows a random path until it encounters recorded measurements again, whilst the 95% credible intervals become wide.

Fifth and Sixth Haringey models (RW1: jags.mod.haringey5; RW2: jags.mod.haringey6) with informative prior distributions from Heathrow posterior distributions

The fifth Haringey model (jags.mod.haringey5) is a random walk of order 1 (RW1) and included as output the parameters: $sigma^2_{25}$, $sigma^2_{26}$ and all $Z2_t$. The model is built using informative prior distributions for the precision (tau_i) that use the mean and variance of the posterior $sigma^2_i$ estimates from the first Heathrow model (jags.mod.heathrow). These estimates have been transformed into precision estimates and applied to the precision estimates (tau_i) for this fifth (RW1) Haringey model using a normal distribution, and are given by:

$$tau25 \sim dnorm(0.036, 0.019) \quad \# \textit{ normal measurement error precision model}$$
$$sigma2_{25} = 1/tau25 \quad \# \textit{ measurement error variance}$$
$$tau26 \sim dnorm(0.040, 0.040) \quad \# \textit{ normal estimate error precision model}$$
$$sigma2_{26} = 1/tau26 \quad \# \textit{ estimate error variance}$$

The sixth Haringey model (jags.mod.haringey6) is a random walk of order 2 (RW2) and included as output the parameters: $sigma^2_{27}$, $sigma^2_{28}$ and all $Z2_t$. The model is built using informative prior distributions for the precision (tau_i) that use the mean and variance of the posterior $sigma^2_i$ estimates from the second Heathrow model (jags.mod.heathrow2). These estimates have been transformed into precision estimates and applied to the precision estimates (tau_i) for this sixth (RW2) Haringey model using a normal distribution, and are given by:

$$tau27 \sim dnorm(0.003, 0.0001) \quad \# \textit{ normal measurement error precision model}$$
$$sigma2_{27} = 1/tau27 \quad \# \textit{ measurement error variance}$$
$$tau28 \sim dnorm(21, 100) \quad \# \textit{ normal estimate error precision model}$$
$$sigma2_{28} = 1/tau28 \quad \# \textit{ estimate error variance}$$

Upon building these models and generating the various density plots, traceplots and tables, it was discovered that the results are no different in terms of which parameters converge compared to the first and second Haringey models (RW1: jags.mod.haringey; RW2: jags.mod.haringey2). Further, the output plots and tables from both sets of models are remarkably similar. Therefore, this suggests that the informative prior distributions from the first and second Heathrow models (RW1: jags.mod.heathrow; RW2: jags.mod.heathrow2) have caused no improvements in either model fit.

Results comparison for Random Walk analyses and Bayesian Inference for Heathrow data

Discussion of how well the chains of each model converged for Y_t estimates when there are recorded measurements ($Heathrow_t$) and missing data are described with the results and explained using the figures therein.

Smoothing effects of Heathrow models (RW1: jags.mod.heathrow; and RW2: jags.mod.heathrow2)

The first Heathrow model (RW1: jags.mod.heathrow) has recorded measurements and Y_t estimates that are generally similar to each other and the Heathrow measurements (Heathrowt) are generally within the 95% credible intervals. Further, these credible intervals are tightly constrained to the Y_t estimates. However, where there are missing data, the Y_t estimates deviate from the mean value of the Heathrow measurements (mean.heathrow) and the credible interval expands greatly, so much so that the lower interval boundary is in negative numbers. Of course, such numbers have no meaning as all measurements must be > 0 (i.e. positive). Once there are measurements to constrain the Y_t estimates, these estimates track closely to the recorded measurements ($Heathrow_t$) and the credible intervals move to be close to the estimates once more. This means that the RW1 model is a very well fitted model to the Heathrow data, but may be overfitted if applied to another dataset (i.e. $Haringey_t$).

In contrast, the second Heathrow model (RW2: jags.mod.heathrow2) has recorded measurements and Y estimates that are generally different to each other and the Heathrow measurements ($Heathrow_t$) are generally outside the 95% credible intervals. Further, these credible intervals are loosely constrained to the Y_t estimates. Further, the Y_t estimates show no less variance with recorded measurements ($Heathrow_t$) and missing data. This means the RW2 model is a poorly fitted model to the Heathrow dataset, but is much smoother than the RW1 model, so may be more applicable to other datasets (i.e. $Haringey_t$) to pick up the general trends.

Lack of convergence in the second Heathrow model (RW2: jags.mod.heathrow2)

The two models can be compared and thought of using the ideas of “degrees of freedom”. With Random Walk 1 modelling, the estimates are based on the original measurements, so there is something like one degree of "freedom" or separation for each chain when plotted in a traceplot. With random walk 2 modelling, estimates are

based on previous estimates, that themselves are based on the measurements. This provides something like two degrees of "freedom" or separation, so the chains do not converge on a traceplot as there is less to constrain them.

Ability to forecast using Heathrow models and relationship with root mean squared errors (RW1: jags.mod.heathrow3 and jags.mod.heathrow7; RW2: jags.mod.heathrow4 and jags.mod.heathrow8)
When comparing the $Y2_t$ estimates with the recorded measurements ($Heathrow_t$) for RW1 (jags.mod.heathrow3), where there are data, root mean squared error (RMSE) values from the fifth Heathrow model (jags.mod.heathrow7), have some variation but always remain above 0.02 and typically below 0.2. The 95% credible interval also has a generally negative lower interval boundary, which is not possible, and an upper interval boundary that typically does not get above 0.5. However, where there are no data, the RMSE value ($Y.rmse_t$) remains relatively constant around a value of 0.09, with a constant credible interval (-0.05, 0.28). This means that when forecasting, with no data to constrain, the forecasted values will remain constant. This may be because their RMSE values ($Y.rmse_t$) tend towards a constant value based on previous estimates and measurements. However, as the RMSE has a large but constant credible interval the credible interval for the forecasted values will grow at a constant rate, as shown in Figure 19.

When comparing the $Y2_t$ estimates with the actual measurements (Heathrow) for RW2 (jags.mod.heathrow4), where there are data the root mean squared error (RMSE) values from jags.mod.heathrow8, have some variation but always remain above 0.1 and typically below 0.75. The 95% credible interval also sometimes has a negative lower interval boundary, which is not possible, and an upper interval boundary that typically remains below 1. However, where there are no data, the RMSE value ($Y.rmse_t$) remains relatively constant around a value of 0.28, with a constant credible interval (-0.15, 0.87). This means that when forecasting, with no data to constrain, the forecasted values will remain relatively stable. This may be because their RMSE values ($Y.rmse_t$) tend towards a constant value based on previous estimates and measurements. However, as the RMSE has a large but constant credible interval the credible interval for the forecasted values will grow at a constant rate, as shown in Figure 22.

Results comparison for Random Walk analyses and Bayesian Inference for Haringey data

Discussion of how well the chains of each model converged for Z_t estimates when there are recorded measurements ($Haringey_t$) and missing data are described with the results and explained using the figures therein.

Use of informative prior distributions on the Haringey models (RW1: jags.mod.haringey3 and jags.mod.haringey5; RW2: jags.mod.haringey4 and jags.mod.haringey6)
There is little or no improvement in which estimates converged for jags.mod.haringey (using non-informative priors) and jags.mod.haringey3 (using informative priors), which both employ RW1 models. The chains converge in traceplots when associated with recorded measurements (Haringeyt) and don't converge when associated with missing data. Still, jags.mod.haringey3 has a lower DIC value (= 19355) than for jags.mod.haringey (= 41485), which suggests the former model with informative priors is a better fit to the data. In addition, there is an improvement in which estimates converged for jags.mod.haringey4 (using informative priors) compared to jags.mod.haringey2 (using non-informative priors), which both employ RW2 models. Also, jags.mod.haringey4 has a lower DIC value (= 8300.2) than for jags.mod.haringey2 (= 13142.6). Output details for all these models can be found earlier in this report.

By following this technique, an update has been made that has produced better fitting prior distributions from previous knowledge of the same model. Of course, this would produce a better model fit for the Haringey data however this improvement may not be applicable in another place. One way to test this is if the posterior distributions from the third and fourth Haringey models were used to provide informative prior distributions for the modelling of the Heathrow data ($Heathrow_t$).

In comparison, using informative prior distributions from the first and second Heathrow models (RW1: jags.mod.heathrow; RW2: jags.mod.heathrow2) have caused no improvements in the model fits compared to the first and second Haringey models (RW1: jags.mod.haringey; RW2: jags.mod.haringey2). This suggests that the situation at Heathrow is different from that in Haringey. This can be simply explained as Heathrow is a major international airport with a constant source of air pollution from aeroplanes that take off and land most hours of the day and night almost every day, whilst Haringey is a suburb in north London with differing levels of road and rail traffic depending on the day and time of day. Perhaps, the Haringey posterior distributions could be used as informative prior distributions for sites either elsewhere in London suburbs or in the suburbs of other UK cities. In contrast, the Heathrow posterior distributions could be used as informative prior distributions for other UK airports.

Conclusions

PM10 air pollution data have been collected at sites in both Heathrow and Haringey, in London between 2000 and 2004, inclusive. These datasets include missing data, with two long periods missing in the Heathrow data between November 2000 and April 2001 as well as between January 2003 and May 2003, and also one long period missing in the Haringey data between September 2002 and April 2003.

Due to the nature of the time series datasets, random walk models of orders 1 and 2, each with 2 chains, 10000 iterations and 5000 iterations "burned", were set up so that the recorded measurements between 2000 and 2003, inclusive, for both sites could be related to "true" estimates of the data. Bayesian inference was performed on each model using a package called JAGS. In this way, the "true" estimates of the data, the root mean square error (RMSE), the model deviance and also the variances of both the measurement error and the estimate error were taken as parameters that JAGS produced as output in traceplots, density plots and data tables. Plots showing the recorded measurements together with the "true" estimates of the data and its 95% credible interval are also produced for each model output.

From application of random walk 1 (RW1) models on both the Heathrow and Haringey data, the resulting traceplots showed that the chains converged when the "true" estimates of the data were associated with recorded measurements for each dataset and didn't converge when associated with missing data. The deviance and both variances showed poor convergence in traceplots. In contrast, the traceplots of the RMSE show that with this parameter the chains converged when the "true" estimates of the data were associated with both recorded measurements and missing data. In the plots of recorded measurements together with the "true" estimates of the data and its 95% credible interval, the "true" estimates fit well with the "noise" in the recorded measurements, with recorded measurements generally within its 95% credible interval. However, where there are missing data, the "true" estimate roams randomly in the time series and its 95% credible interval enlargens greatly until there are recorded measurements again in the time series.

From application of random walk 2 (RW2) models on both the Heathrow and Haringey data, the resulting traceplots showed that the chains did not converge when the "true" estimates of the data were associated with either recorded measurements or missing data for each dataset. The deviance and both variances also showed very poor convergence in traceplots. In contrast, the traceplots of the RMSE show that with this parameter the chains converged when the "true" estimates of the data were associated with missing data for each dataset and didn't converge when associated with recorded measurements. In the plots of recorded measurements together with the "true" estimates of the data and its 95% credible interval, the "true" estimates do not fit well with the "noise" in the recorded measurements and lie often below these data. The 95% credible interval of the "true" estimate varies with little relationship to the recorded measurements. Also, the "true" estimate continues to roam in a similar manner as when recorded measurements are associated and its 95% credible interval shows no significant change in size.

Forecasting was performed on the Heathrow data using both random walk models for the first week of 2004, with the recorded measurements deliberately coerced into missing data, so that they did not influence the model outputs. With a RW1 model, the forecasted values remained constant during the first week of 2004, however the 95% credible interval for the forecasted values grew at a constant rate. With a RW2 model, the forecasted values remained relatively stable, however the 95% credible interval for the forecasted values grew at a constant rate.

Lastly, the posterior distributions from both random walk models on both the Heathrow and Haringey datasets were used to produce new models with informative prior distributions to perform Bayesian inference on the Haringey dataset. With the informative prior distributions from the posterior distributions in the Haringey RW models, there was little improvement in the model fit for the new RW1 model, however there was significant improvement in the model fit for the new RW2 model. This involved traceplots that showed that the chains had almost converged when the "true" estimates of the data were associated with recorded measurements for each dataset but still didn't converge when associated with missing data. Also, the deviance and both variances showed poor but improved convergence in traceplots. With the informative prior distributions from the posterior distributions in the Heathrow RW models, there was little improvement in the model fit for both the new RW1 and RW2 models.

Appendix – Code for Bayesian inference

```
## A. Bayesian Inference [80 marks]

# The dataset contains measurements of particulate matter (PM10) air pollution in London (measured at
# the Heathrow and Haringey sites) for 2000 to 2004. The data can be found in London_Pollution.csv.

## Install and load packages to run London air pollution data
library(Rmisc)
library(tidyverse)
#library(TailRank)
library(R2jags)
library(coda)
library(lattice)
#library(jagsplot)
library(MCMCvis)
library(rjags)
#library(mcmcplots)
library(readr)
library(spdep)
library(sf)
library(CARBayes)
library(rgdal)
library(rgeos)
library(rdist)
library(gridExtra)
library(leaflet)
library(rnrfa)
library(maptools)
library(ggmap)
library(sp)
library(dplyr)
library(kableExtra)
library(matrixNormal)


## Reading in London pollution data
london_pollution <- read.csv("London_Pollution.csv")
london_pollution$Date <- as.Date(london_pollution$Date, format = "%d/%m/%Y")
head(london_pollution)
tail(london_pollution)
london_pollution[350:400,]
london_pollution[50:100,]

## The date and heathrow air pollution data are in the 2nd and 3rd columns, respectively,
## and there are 1827 measurements from 1st January 2000 to 31st December 2004:
heathrow <- dplyr::select(london_pollution, Date, Heathrow)
head(heathrow)
tail(heathrow)
count(heathrow)
sapply(heathrow, class)

## The date and haringey air pollution data are in the 2nd and 4th columns, respectively,
## and there are 1827 measurements from 1st January 2000 to 31st December 2004:
haringey <- dplyr::select(london_pollution, Date, Haringey)
head(haringey)
tail(haringey)
```

```
count(haringey)
sapply(haringey, class)

## 1. [4 marks] Summarise the two sets of data and calculate the number of missing data points for each
## monitoring location, by year. Comment on whether the patterns of missingness have changed over time.

summary(heathrow[,2])
summary.heathrow <- data.frame(summary.heathrow = unclass(summary(heathrow[,2])),
                               check.names = FALSE, stringsAsFactors = FALSE)
kbl(summary.heathrow) %>% kable_styling()
# Min.  1st Qu.  Median   Mean  3rd Qu.   Max.    NA's
# 3.10   15.50   19.50   21.72   25.90   69.90    404

summary(haringey[,2])
summary.haringey <- data.frame(summary.haringey = unclass(summary(haringey[,2])),
                               check.names = FALSE, stringsAsFactors = FALSE)
kbl(summary.haringey) %>% kable_styling()
# Min.  1st Qu.  Median   Mean  3rd Qu.    Max.    NA's
# 5.90   14.80   18.50   20.24   23.60    65.90    278


# Creating new Day, Month and Year columns from Date column for Heathrow data
heathrow <- heathrow %>% separate(Date, into = c('Year', 'Month', 'Day'))
heathrow2 <- cbind(heathrow, Date = london_pollution$Date)
head(heathrow2)
# arrange columns with Date, Day, Month, Year, Heathrow air pollution
heathrow2 <- heathrow2 %>% select(Date, Year, Month, Day, Heathrow)
head(heathrow2)
# filter heathrow2 data frame for rows with NAs in Heathrow column
heathrow_NA <- heathrow2 %>% filter(is.na(Heathrow))
# select just the Year and Month columns
heathrow_by_month <- heathrow_NA %>% select(-Date, -Day, -Heathrow)
head(heathrow_by_month)
# report frequency of NAs per month
# heathrow_NA_freq <- aggregate(list(Frequency = rep(1, nrow(heathrow_by_month))), heathrow_by_month,
length)
# head(heathrow_NA_freq)

# plot NA Frequency as a barplot for each month by year
ggplot(data = heathrow_by_month) +
  geom_bar(mapping = aes(x = Month, fill = Year))
# plot NA Frequency as facetted barplots for each month by year
ggplot(data = heathrow_by_month) +
  geom_bar(mapping = aes(x = Month, fill = Year)) +
  facet_wrap(~ Year)

# The facet wrap of Frequency of NA results by month in Heathrow shows that there were very few missing
# results during January-October 2000, with 6 in May, 5 in October and only two or less in any other month.
# However, there were no results recorded during November-December 2000 and this lack of results continued
# throughout January-April 2001. During the rest of 2001, there are 8 missing results in April, 12 in October
# and only five or less in any other month. During 2002, there were only four or less in any month.
# During 2003, all results are missing for January-May and there are 12 missing results during June.
# Following this, there are no more than 3 missing results for any month thereafter during 2003 and 2004.

# The long periods of time where no data are recorded probably represents either equipment maintainance and/or
upgrades.
```

```
# The other times of missing results may represent days when either there was a power cut or someone forgot to collect the results.


# Creating new Day, Month and Year columns from Date column for Haringey data
haringey <- haringey %>% separate(Date, into = c('Year', 'Month', 'Day'))
haringey2 <- cbind(haringey, Date = london_pollution$Date)
head(haringey2)
# arrange columns with Date, Day, Month, Year, Haringey air pollution
haringey2 <- haringey2 %>% select(Date, Year, Month, Day, Haringey)
head(haringey2)
# filter haringey2 data frame for rows with NAs in Haringey column
haringey_NA <- haringey2 %>% filter(is.na(Haringey))
# select just the Year and Month columns
haringey_by_month <- haringey_NA %>% select(-Date, -Day, -Haringey)
head(haringey_by_month)
# report frequency of NAs per month
# haringey_NA_freq <- aggregate(list(Frequency = rep(1, nrow(haringey_by_month))), haringey_by_month, length)
# head(haringey_NA_freq)

# plot NA Frequency as a barplot for each month by year
ggplot(data = haringey_by_month) +
  geom_bar(mapping = aes(x = Month, fill = Year))
# plot NA Frequency as facetted barplots for each month by year
ggplot(data = haringey_by_month) +
  geom_bar(mapping = aes(x = Month, fill = Year)) +
  facet_wrap(~ Year)

# The facet wrap of Frequency of NA results by month in Haringey shows that there were very few missing
# results during 2000, with 10 in March and 9 in May, and only two or less in any other month. During 2001,
# there were only 2 missing results in total, which was during February. During 2002, there were only 5
# missing results during January-August, however all the results are missing form September to December.
# During 2003, the missing results continue through out January-March and most of April. Following this,
# there are no more than 3 missing results for any month thereafter during 2003 and 2004.

# The long periods of time where no data are recorded probably represents either equipment maintainance and/or upgrades.
# The other times of missing results may represent days when either there was a power cut or someone forgot to collect the results.


## 2. [3 marks] Plot the PM10 measurements against time for the two sites, highlighting (showing clearly)
## the periods of missing data.

# Creating new Day, Month and Year columns from Date column for Heathrow data
heathrow <- heathrow %>% separate(Date, into = c('Year', 'Month', 'Day'))
heathrow2 <- cbind(heathrow, Date = london_pollution$Date)
head(heathrow2)
# arrange columns with Date, Day, Month, Year, Heathrow air pollution
heathrow2 <- heathrow2 %>% dplyr::select(Date, Year, Month, Day, Heathrow)
head(heathrow2)
# Convert all NA data to zeros
heathrow2[is.na(heathrow2)] <- 0

# Separate Heathrow data into data frames for each year and make NAs = 0
# 2000
heathrow_2000 <- heathrow2 %>% filter(Year == "2000")
```

```
head(heathrow_2000)
tail(heathrow_2000)
# 2001
heathrow_2001 <- heathrow2 %>% filter(Year == "2001")
head(heathrow_2001)
tail(heathrow_2001)
# 2002
heathrow_2002 <- heathrow2 %>% filter(Year == "2002")
head(heathrow_2002)
tail(heathrow_2002)
# 2003
heathrow_2003 <- heathrow2 %>% filter(Year == "2003")
head(heathrow_2003)
tail(heathrow_2003)
# 2004
heathrow_2004 <- heathrow2 %>% filter(Year == "2004")
head(heathrow_2004)
tail(heathrow_2004)

# On each line plot, NAs are represented where the data = 0

# Plot each year of data as a line plot - 2000
ggplot(data = heathrow_2000) +
  geom_line(mapping = aes(x = Date, y = Heathrow)) +
  labs(y = "PM10", title = "PM10 measurements at Heathrow during 2000")
# Plot each year of data as a line plot - 2001
ggplot(data = heathrow_2001) +
  geom_line(mapping = aes(x = Date, y = Heathrow)) +
  labs(y = "PM10", title = "PM10 measurements at Heathrow during 2001")
# Plot each year of data as a line plot - 2002
ggplot(data = heathrow_2002) +
  geom_line(mapping = aes(x = Date, y = Heathrow)) +
  labs(y = "PM10", title = "PM10 measurements at Heathrow during 2002")
# Plot each year of data as a line plot - 2003
ggplot(data = heathrow_2003) +
  geom_line(mapping = aes(x = Date, y = Heathrow)) +
  labs(y = "PM10", title = "PM10 measurements at Heathrow during 2003")
# Plot each year of data as a line plot - 2004
ggplot(data = heathrow_2004) +
  geom_line(mapping = aes(x = Date, y = Heathrow)) +
  labs(y = "PM10", title = "PM10 measurements at Heathrow during 2004")

# Plot data as a line plot - full length
ggplot(data = heathrow2) +
  geom_line(mapping = aes(x = Date, y = Heathrow)) +
  labs(y = "PM10", title = "PM10 measurements at Heathrow between 2000-2004")




# Creating new Day, Month and Year columns from Date column for Haringey data
haringey <- haringey %>% separate(Date, into = c('Year', 'Month', 'Day'))
haringey2 <- cbind(haringey, Date = london_pollution$Date)
head(haringey2)
# arrange columns with Date, Day, Month, Year, Haringey air pollution
haringey2 <- haringey2 %>% dplyr::select(Date, Year, Month, Day, Haringey)
head(haringey2)
```

```
# Convert all NA data to zeros
haringey2[is.na(haringey2)] <- 0

# Separate Haringey data into data frames for each year and make NAs = 0
# 2000
haringey_2000 <- haringey2 %>% filter(Year == "2000")
head(haringey_2000)
tail(haringey_2000)
# 2001
haringey_2001 <- haringey2 %>% filter(Year == "2001")
head(haringey_2001)
tail(haringey_2001)
# 2002
haringey_2002 <- haringey2 %>% filter(Year == "2002")
head(haringey_2002)
tail(haringey_2002)
# 2003
haringey_2003 <- haringey2 %>% filter(Year == "2003")
head(haringey_2003)
tail(haringey_2003)
# 2004
haringey_2004 <- haringey2 %>% filter(Year == "2004")
head(haringey_2004)
tail(haringey_2004)

# On each line plot, NAs are represented where the data = 0

# Plot each year of data as a line plot - 2000
ggplot(data = haringey_2000) +
  geom_line(mapping = aes(x = Date, y = Haringey)) +
  labs(y = "PM10", title = "PM10 measurements at Haringey during 2000")
# Plot each year of data as a line plot - 2001
ggplot(data = haringey_2001) +
  geom_line(mapping = aes(x = Date, y = Haringey)) +
  labs(y = "PM10", title = "PM10 measurements at Haringey during 2001")
# Plot each year of data as a line plot - 2002
ggplot(data = haringey_2002) +
  geom_line(mapping = aes(x = Date, y = Haringey)) +
  labs(y = "PM10", title = "PM10 measurements at Haringey during 2002")
# Plot each year of data as a line plot - 2003
ggplot(data = haringey_2003) +
  geom_line(mapping = aes(x = Date, y = Haringey)) +
  labs(y = "PM10", title = "PM10 measurements at Haringey during 2003")
# Plot each year of data as a line plot - 2004
ggplot(data = haringey_2004) +
  geom_line(mapping = aes(x = Date, y = Haringey)) +
  labs(y = "PM10", title = "PM10 measurements at Haringey during 2004")

# Plot data as a line plot - full length
ggplot(data = haringey2) +
  geom_line(mapping = aes(x = Date, y = Haringey)) +
  labs(y = "PM10", title = "PM10 measurements at Haringey between 2000-2004")
```

3. [5 marks] The locations in Eastings and Northings of the two locations are Heathrow: (508399, 176744);
and Haringey: (533890, 190638). Plot these two monitor locations on a map of London and comment on any

```
## difference you found in the summaries of the data in the context of the geographical location of the
## monitoring sites. The necessary shapefiles are on the ELE page of the course.

# Reading in London shapefiles
London <- readOGR(dsn = '.', layer = 'London')
plot(London)
# SET the CRS of the object
proj4string(London) <- CRS("+init=epsg:27700")
# NOW we can transform to lat/lon
Londonxy <- spTransform(London, CRS("+proj=longlat +ellps=WGS84 +datum=WGS84"))

# and finally, leaflet will accept this spatial object
new %>% leaflet()

wgs84 = '+proj=longlat +datum=WGS84'
Londonxy <- spTransform(London, CRS(wgs84))

## Add grid references to a data frame ##
Location <- c("Heathrow", "Haringey")
Eastings <- c(508399, 533890)
Northings <- c(176744, 190630)
Sites <- data.frame(Location, Eastings, Northings)

# Linking to GEOS 3.6.1, GDAL 2.2.3, proj.4 4.9.3
London_Sites <- Sites %>%
  st_as_sf(coords = c("Eastings", "Northings"), crs = 27700) %>%
  st_transform(4326) %>% st_coordinates() %>% as_tibble()
head(London_Sites)

# Add default OpenStreetMap map tiles and print the map at zoom = 10
London_map <- leaflet(London_Sites) %>% addTiles() %>%
  setView(lng = -0.1, lat = 51.5, zoom = 10) %>%
  addPolygons(data = Londonxy, weight = 3, col = 'red') %>%
  addCircleMarkers(lng = London_Sites$X, lat = London_Sites$Y,
                   radius = 3, opacity = 1, label = Sites$Location)
London_map


# Considering the Heathrow data, there is missing data. We are going to fit a model that allows us to estimate
# these missing data by treating them as model parameters that will be estimated (and we find posterior
# distributions for them). As we have time series data, we are going to use the fact that day-to-day measurements
# will be correlated, i.e. today's measurement will correlate with yesterday's.

# A random walk process of order 1, RW(1), is defined at time t as:
# Yt − Yt−1 = wt
# Yt = Yt−1 + wt
# Where wt are a set of realisations of random (or white) noise, e.g. wt ∼ N(0, σw^2).
# Note the first line refers to the differences in the values at consecutive time points being white noise.

# We are interested in fitting a random walk model to the Heathrow data (Heathrow).
# The model will be of the following form:
# Heathrowt ∼ N(Yt, σv^2)
# Yt ∼ N(Yt−1, σw^2)
# Where σw^2 is the variance of the white noise process associated to the random walk. We then make noisy
# measurements of this random walk process, thus Heathrowt, the measurement we have at time t, equals
# the true value of the underlying process Yt plus some measurement error. In the formula above, σv2
# is the variance of this measurement error.
```

4. [16 marks] Code this model in JAGS to analyse the Heathrow data from 1st January 2000 to 31st December 2003
(NOTE the end year). Hint: due to the nature of the model you will have to explicitly specify a value for Y1
in the model (i.e. for the first time point as Y0 doesn't exist). One suggestion might be mu ~ dnorm(0, 0.001).
Run the model for 10,000 iterations, with 2 chains, discarding the first 5,000 as 'burn-in'. Produce trace plots
for the chains and summaries for the fitted parameters (including the missing data). Hint: You should initialise
both chains. One suggestion might be using the mean and median to initialise the missing values of Heathrow, and
using random uniforms (with a narrow interval centred around say 20) to initialise Y.

```
## Reading in London pollution data
london_pollution <- read.csv("London_Pollution.csv")
london_pollution$Date <- as.Date(london_pollution$Date, format = "%d/%m/%Y")
head(london_pollution)
tail(london_pollution)
london_pollution[350:400,]
london_pollution[50:100,]

## The date and heathrow air pollution data are in the 2nd and 3rd columns, respectively,
## and there are 1827 measurements from 1st January 2000 to 31st December 2004:
heathrow <- dplyr::select(london_pollution, Date, Heathrow)
head(heathrow)
tail(heathrow)
count(heathrow)
sapply(heathrow, class)

# Creating new Day, Month and Year columns from Date column for Heathrow data
heathrow <- heathrow %>% separate(Date, into = c('Year', 'Month', 'Day'))
heathrow2 <- cbind(heathrow, Date = london_pollution$Date)
head(heathrow2)
# arrange columns with Date, Day, Month, Year, Heathrow air pollution
heathrow2 <- heathrow2 %>% dplyr::select(Date, Year, Month, Day, Heathrow)
head(heathrow2)
# heathrow data for just 2000-2003
heathrow3 <- heathrow2 %>% dplyr::filter(Year != 2004)
tail(heathrow3)
heathrow3[1001:1100,]
heathrow3[1101:1200,]
heathrow3[1201:1300,]

# Set seed for reproducibility
set.seed(234)
# Set N to be the length of the data
N1 <- length(heathrow3$Heathrow)
N1
# Mean of Heathrow data
mean.heathrow <- mean(heathrow3$Heathrow, na.rm = TRUE)
mean.heathrow
# Extract Heathrow_centred column data
Heathrow <- heathrow3$Heathrow
Heathrow


# List the data to be used
jags.data.heathrow <- list("Heathrow", "N1")
```

```
# Model
jags.mod.heathrow <- function(){
  Y[1] ~ dnorm(0, 0.001)
  for(t in 2:N1){
    Heathrow[t] ~ dnorm(Y[t], tau1)            # centred normal likelihood of heathrow data
    Y[t] ~ dnorm(Y[t-1], tau2)                 # normal prediction model
  }
  # priors on measurement error and white noise variances
  tau1 ~ dgamma(0.001, 0.001)                  # gamma measurement error precision model
  sigma2_1 <- 1/tau1                           # measurement error variance
  tau2 ~ dgamma(0.001, 0.001)                  # gamma estimate error precision model
  sigma2_2 <- 1/tau2                           # estimate error variance
}

# Specify initial values
inits.heathrow1 <- list("Y[1]" = 22, "is.na(Heathrow)" = mean.heathrow)
inits.heathrow2 <- list("Y[1]" = 20, "is.na(Heathrow)" = mean.heathrow)
jags.inits.heathrow <- list(inits.heathrow1, inits.heathrow2)

# Monitor the parameters to be used for the prediction
jags.param.heathrow <- c("sigma2_1","sigma2_2","Y")

# Fitting the new model
jags.mod.fit.heathrow <- jags(data = jags.data.heathrow, inits = jags.inits.heathrow,
                      parameters.to.save = jags.param.heathrow, n.chains = 2,
                      n.iter = 10000, n.burnin = 5000, n.thin = 1,
                      model.file = jags.mod.heathrow)

# Get point and interval estimates
print(jags.mod.fit.heathrow)
# Inference for Bugs model at
"/var/folders/pc/hksslngn6_56y6dc8464zthm0000gn/T//RtmplXgoVU/model28a1bd5e5aa.txt", fit using jags,
# 2 chains, each with 10000 iterations (first 5000 discarded)
# n.sims = 10000 iterations saved
#         mu.vect sd.vect    2.5%     25%     50%     75%   97.5%  Rhat n.eff
# Y[1]     10.549   6.254  -1.459   6.295  10.541  14.773  22.880 1.002  1000
# Y[2]     10.946   3.990   3.182   8.235  10.914  13.617  18.845 1.001  3000
# Y[3]     10.904   3.512   4.033   8.554  10.873  13.272  17.801 1.003   890
# Y[4]     13.001   3.406   6.185  10.698  13.023  15.263  19.695 1.002  1300
# Y[5]     16.991   3.410  10.098  14.741  17.019  19.309  23.571 1.001  3200
# Y[6]     18.511   3.455  11.882  16.167  18.486  20.808  25.402 1.001 10000
# Y[7]     20.105   3.435  13.321  17.807  20.157  22.396  26.759 1.001 10000
# Y[8]     21.155   3.424  14.288  18.859  21.207  23.455  27.838 1.002  2300
# Y[9]     26.801   3.577  19.623  24.505  26.833  29.199  33.614 1.002  1000
# Y[10]    29.800   3.972  21.871  27.262  29.837  32.481  37.319 1.005   330    1# ...
# For each parameter, n.eff is a crude measure of effective sample size,
# and Rhat is the potential scale reduction factor (at convergence, Rhat=1).

# DIC info (using the rule, pD = var(deviance)/2)
# pD = 17959.9 and DIC = 24459.2
# DIC is an estimate of expected predictive error (lower deviance is better).


# plot density manually
sim.values.heathrow <- jags.mod.fit.heathrow$BUGSoutput$sims.list
kbl(jags.mod.fit.heathrow$BUGSoutput$sims.list) %>% kable_styling()
```

```
df.heathrow <- data.frame(sigma2_1 = sim.values.heathrow$sigma2_1,
                          sigma2_2 = sim.values.heathrow$sigma2_2)

heathrow_sigma2_1 <- ggplot(data = df.heathrow, aes(x = sigma2_1)) +
  geom_density(colour = '#0093af', fill = '#0093af', alpha = 0.3) +
  theme(axis.title = element_blank(),
        axis.text = element_text(size = 12),
        plot.title = element_text(size = 14)) +
  geom_vline(xintercept = 0, linetype = "dashed",
             colour = "#aa0078", size = 1) +
  ggtitle('Posterior density of sigma2_1 - measurement variance')
heathrow_sigma2_1

heathrow_sigma2_2 <- ggplot(data = df.heathrow, aes(x = sigma2_2)) +
  geom_density(colour = '#0093af', fill = '#0093af', alpha = 0.3) +
  theme(axis.title = element_blank(),
        axis.text = element_text(size = 12),
        plot.title = element_text(size = 14)) +
  geom_vline(xintercept = 0, linetype = "dashed",
             colour = "#aa0078", size = 1) +
  ggtitle('Posterior density of sigma2_2 - estimate variance')
heathrow_sigma2_2



## 5. [3 marks] Comment on whether the chains for all the parameters have converged.

# Create an MCMC object from the output of the heathrow model
jags.mcmc.heathrow <- as.mcmc(jags.mod.fit.heathrow)
# Graphical parameters
par(mar = c(2,4,4,2), cex = 1.0)
layout(matrix(c(1,1,2,2),2,2,byrow=TRUE))
# for a traceplot, producing for groups of 100 traceplots
traceplot(jags.mcmc.heathrow[,101:150], params = c("sigma2_1", "sigma2_2", "Y"))
traceplot(jags.mcmc.heathrow[,251:300], params = c("sigma2_1", "sigma2_2", "Y"))
traceplot(jags.mcmc.heathrow[,1:4], params = c("sigma2_1", "sigma2_2", "Y"))

# Trace plots were generated from jags.mcmc.heathrow for each parameter (sigma2_1, sigma2_2,
# deviance, Y[1] and selected examples of Y that covered both where measurements are known and
# where they are missing). The trace plots for deviance, sigma2_1 and sigma2_2 are very close
# to converging for each chain however there is a small amount deviation from each other.
# For the Y estimates, the amount of convergence relies greatly on whether the parameter estimate
# has associated measurements in the Heathrow dataset or are missing values. The chains converge
# if the Y estimate has an associated Heathrow measurement and do not converge if there are missing
# values. However, the amount that a Y estimate does not converge becomes greater if the estimate
# is further from a Y estimate that has an associated measurement.

# In this way, Y estimates 1100-1259 are missing data, but the convergence of Y estimate 1150 is
# worse than Y estimate 1110, which is worse than Y estimate 1100. Conversely, the convergence of
# Y estimate 1265 (with ass. measurement) is better than Y estimate 1260 (with ass. measurement),
# which is better than Y estimate 1259 (missing data), which again is better than Y estimate 1255
# (missing data). This shows that as each estimate is based on the one previous, there is a "memory"
# of previous Y estimates that reduces the convergence where estimates "go further into" blocks
# missing data and improves the convergence as the estimates become associated with measurements once more.


# for summary statistics
```

```
summary(jags.mcmc.heathrow)
# Iterations = 5001:10000
# Thinning interval = 1
# Number of chains = 2
# Sample size per chain = 5000

# 1. Empirical mean and standard deviation for each variable, plus standard error of the mean:
#             Mean       SD  Naive SE Time-series SE
# deviance 6499.35316 192.980  1.92980      15.25564
# sigma2_1   27.53398   7.957  0.07957       0.60604
# sigma2_2   25.01324   4.971  0.04971       0.45576
# Y[1]       10.54893   6.254  0.06254       0.10093
# Parameters with associated measurements
# Y[1150]    20.443    29.317  0.29317       8.03162
# Y[1151]    20.373    29.249  0.29249       7.35802
# Y[1152]    20.429    29.182  0.29182       6.81700
# Y[1153]    20.552    29.057  0.29057       7.08630
# Y[1154]    20.892    28.931  0.28931       6.66124
# Y[1155]    21.211    28.853  0.28853       7.23935
# Y[1156]    21.525    28.630  0.28630       7.67131
# Y[1157]    21.837    28.415  0.28415       7.01433
# Y[1158]    22.159    28.123  0.28123       6.55003
# Y[1159]    22.394    27.776  0.27776       6.47392
# Parameters with missing data
# Y[1160]    22.487    27.412  0.27412       6.04469
# Y[1161]    22.465    27.120  0.27120       6.03380
# Y[1162]    22.335    26.883  0.26883       6.04510
# Y[1163]    22.151    26.652  0.26652       6.32909
# Y[1164]    22.062    26.550  0.26550       6.13987
# Y[1165]    22.047    26.494  0.26494       5.82835
# Y[1166]    22.004    26.432  0.26432       6.16386
# Y[1167]    21.988    26.286  0.26286       5.77716
# Y[1168]    21.966    26.209  0.26209       6.09513
# Y[1169]    21.792    26.041  0.26041       5.85387

# 2. Quantiles for each variable:
#             2.5%       25%       50%       75%     97.5%
# deviance 6161.6942 6399.13995 6496.6865 6590.798 6796.42
# sigma2_1   19.3429   24.36541   26.8355   29.521   36.42
# sigma2_2   15.5689   22.10253   25.0493   27.971   34.64
# Y[1]       -1.4593    6.29525   10.5412   14.773   22.88
# Parameters with associated measurements
# Y[1150]   -33.59671 -2.370e+00   21.222   43.586   72.18
# Y[1151]   -33.81237 -2.334e+00   21.343   43.498   71.73
# Y[1152]   -33.96751 -2.258e+00   21.473   43.783   71.29
# Y[1153]   -33.43377 -2.313e+00   21.517   44.156   70.58
# Y[1154]   -32.14822 -1.523e+00   21.826   44.535   70.40
# Y[1155]   -31.25195 -1.484e+00   22.701   44.816   69.75
# Y[1156]   -30.68705 -1.117e+00   23.283   44.695   69.42
# Y[1157]   -30.58907 -7.987e-01   24.356   45.086   69.42
# Y[1158]   -30.31852 -2.916e-01   24.726   44.710   68.97
# Y[1159]   -29.91920  1.602e-01   24.978   44.473   68.87
# Parameters with missing data
# Y[1160]   -29.91471  7.846e-01   25.067   43.959   68.30
# Y[1161]   -30.02804  1.228e+00   25.149   43.513   67.60
# Y[1162]   -29.72023  1.345e+00   25.176   43.116   67.06
# Y[1163]   -29.29676  1.474e+00   25.044   42.841   66.70
```

```
# Y[1164]   -28.08269  1.476e+00   24.831   42.463   66.25
# Y[1165]   -28.13755  1.353e+00   24.511   42.407   66.80
# Y[1166]   -27.99780  1.082e+00   24.611   42.024   67.04
# Y[1167]   -26.54037  1.254e+00   24.504   41.809   67.85
# Y[1168]   -26.19140  1.358e+00   24.038   41.647   68.35
# Y[1169]   -25.57739  1.182e+00   23.452   41.309   68.76



# Produce an MCMC trace of sigma2_1 output
MCMCtrace(jags.mcmc.heathrow,
        params = c('sigma2_1'),
        type = 'density',
        ind = TRUE,
        ISB = FALSE,
        pdf = FALSE,
        col_den = c("#aa0078", "#0093af"))

# Produce an MCMC trace of sigma2_2 output
MCMCtrace(jags.mcmc.heathrow,
        params = c('sigma2_2'),
        type = 'density',
        ind = TRUE,
        ISB = FALSE,
        pdf = FALSE,
        col_den = c("#aa0078", "#0093af"))

# Produce an MCMC trace of Y output
MCMCtrace(jags.mcmc.heathrow,
        params = c('Y'),
        type = 'density',
        ind = TRUE,
        ISB = FALSE,
        pdf = FALSE,
        col_den = c("#aa0078", "#0093af"))

gelman.heathrow <- gelman.diag(jags.mcmc.heathrow)
gelman.heathrow
# Potential scale reduction factors:
#          Point est. Upper C.I.
# deviance       1.00       1.00
# sigma2_1       1.00       1.00
# sigma2_2       1.00       1.01
# Y[1]           1.00       1.00
# Parameters with associated measurements
# Y[1090]        1.00       1.01
# Y[1091]        1.00       1.00
# Y[1092]        1.00       1.00
# Y[1093]        1.00       1.00
# Y[1094]        1.00       1.00
# Y[1095]        1.00       1.00
# Y[1096]        1.00       1.00
# Y[1097]        1.00       1.00
# Y[1098]        1.00       1.00
# Y[1099]        1.00       1.01
# Parameters with missing data
# Y[1100]        1.01       1.04
```

```
# Y[1101]      1.02     1.09
# Y[1102]      1.04     1.17
# Y[1103]      1.07     1.27
# Y[1104]      1.10     1.38
# Y[1105]      1.13     1.47
# Y[1106]      1.16     1.56
# Y[1107]      1.18     1.64
# Y[1108]      1.21     1.73
# Y[1109]      1.24     1.82

# Multivariate psrf
# 2.45

# Create a data.frame from the gelman.diag output to put into a kable table
rhat.heathrow <- as.data.frame((gelman.heathrow)$psrf)
kbl(rhat.heathrow) %>% kable_styling()

# Create a gelman plot of all the parameters and their densities
gelman.plot(jags.mcmc.heathrow)
```

6. [5 marks] Extract the posterior means and 95% credible intervals for Yt, and plot them against time, together
with the original data (the measurements). Comment on the width of the credible interval during the periods of missing data.

```
# Below is the code to plot the Heathrow measurements (NA values removed) against the Y estimates (including where
# missing data has been estimated) and the associated upper and lower 95% credible intervals for the Y estimates.
# Typically, the measurements and the Y estimates are similar to each other and the Heathrow measurements are within
# the 95% credible intervals. Further, these credible intervals are tightly constrained the the Y estimates.
# However, where there are missing data, the Y estimates deviate from the mean value of the Heathrow measurements
# and the credible interval expands greatly, so much so that the lower interval boundary is in negative numbers.
# Of course, such numbers have no meaning as all measurements must be >0 (i.e. positive). Once there are measurements
# to constrain the Y estimates, these estimates track closely to the measurements and the credible intervals move
# to be close to the estimates once more.

# ```{r}
mu.heathrow <- jags.mod.fit.heathrow$BUGSoutput$mean$Y
sd.heathrow <- jags.mod.fit.heathrow$BUGSoutput$sd$Y

# ```{r}
df.heathrow2 <- data.frame(x = heathrow3$Date, y1 = mu.heathrow, y2 = heathrow3$Heathrow,
                           lower.heathrow = mu.heathrow - 1.96*sd.heathrow,
                           upper.heathrow = mu.heathrow + 1.96*sd.heathrow)

ggplot(data = df.heathrow2) +
  geom_point(aes(x = x, y = y1), colour = 'grey31', size = 2) +
  geom_point(aes(x = x, y = y2), colour = 'blue', size = 2) +
  geom_line(aes(x = x, y = y1), colour = '#0093af', size = 1) +
  geom_line(aes(x = x, y = lower.heathrow), linetype = "dashed", colour = "#aa0078", size = 1) +
  geom_line(aes(x = x, y = upper.heathrow), linetype = "dashed", colour = "#aa0078", size = 1) +
  xlab('Date') + ylab('Y (estimates) and Heathrow (measurements) of PM10 at Heathrow') +
  ggtitle('PM10 measurements and estimates at Heathrow for 2000-2003') +
```

```
 theme(axis.title = element_text(size = 14),
       axis.text = element_text(size = 12),
       plot.title = element_text(size = 14))
```

An alternative model is a random walk process of order 2, RW(2). This assumes that the 'differences between differences'
is white noise and is defined at time t as:
(Yt − Yt−1) − (Yt−1 − Yt−2) = wt
Yt = 2Yt − 1 − Yt − 2 + wt
Where again wt are a set of realisations of random (or white) noise, e.g. wt ∼ N (0, σw2).

That is now we are interested in fitting a random walk model of order 2 to the Heathrow data. The model will be of the following form:
Heathrowt ∼ N(Yt, σv^2)
Yt ∼ N(2Yt − 1 − Yt − 2, σw^2)
Again, σw^2 is the variance of the white noise process, and σv^2 is the variance of the measurement error.

7. [12 marks] Code this RW(2) model in JAGS to analyse the Heathrow data from 1st January 2000 to 31st December 2003
(NOTE the end year). Run the model for 10,000 iterations, discarding the first 5,000 as 'burn-in'. Produce trace plots
for the chains and summaries for the fitted parameters (including the missing data). Comment on the differences between
the smoothing effects of the two models. For this you might find it useful to plot the outcome for the first quarter of
2000 separately (for both models). Note that getting this model to converge might be quite tricky. Instead of spending
much time trying to get it to converge, you should instead try to explain why we might see lack of convergence here.

```
# Set seed for reproducibility
set.seed(234)
# Set N to be the length of the data
N1 <- length(heathrow3$Heathrow)
N1
# Mean of Heathrow data
mean.heathrow <- mean(heathrow3$Heathrow, na.rm = TRUE)
mean.heathrow
# Extract Heathrow_centred column data
Heathrow <- heathrow3$Heathrow
Heathrow
```

```
# List the data to be used
jags.data.heathrow2 <- list("Heathrow", "N1")

# Model 2
jags.mod.heathrow2 <- function(){
  Y[1] ~ dnorm(0, 0.001)
  Y[2] ~ dnorm(0, 0.001)
  for(t in 3:N1){
    Heathrow[t] ~ dnorm(Y[t], tau3)            # normal likelihood of heathrow data
    Y[t] ~ dnorm((2*Y[t-1] - Y[t-2]), tau4)    # normal prediction model
```

```
  }
  # priors on measurement error and white noise variances
  tau3 ~ dgamma(0.001, 0.001)          # gamma measurement error precision model
  sigma2_3 <- 1/tau3                   # measurement error variance
  tau4 ~ dgamma(0.001, 0.001)          # gamma estimate error precision model
  sigma2_4 <- 1/tau4                   # estimate error variance
}

# Specify initial values
inits.heathrow3 <- list("Y[1]" = 22, "is.na(Heathrow)" = mean.heathrow)
inits.heathrow4 <- list("Y[1]" = 20, "is.na(Heathrow)" = mean.heathrow)
jags.inits.heathrow2 <- list(inits.heathrow3, inits.heathrow4)

# Monitor the parameters to be used for the prediction
jags.param.heathrow2 <- c("sigma2_3","sigma2_4","Y")

# Fitting the new model
jags.mod.fit.heathrow2 <- jags(data = jags.data.heathrow2, inits = jags.inits.heathrow2,
                    parameters.to.save = jags.param.heathrow2, n.chains = 2,
                    n.iter = 10000, n.burnin = 5000, n.thin = 1,
                    model.file = jags.mod.heathrow2)

# Get point and interval estimates
print(jags.mod.fit.heathrow2)
# Inference for Bugs model at
"/var/folders/pc/hksslngn6_56y6dc8464zthm0000gn/T//RtmpuRFXJq/modelb5665ddda68.txt", fit using jags,
# 2 chains, each with 10000 iterations (first 5000 discarded)
# n.sims = 10000 iterations saved
#         mu.vect sd.vect    2.5%     25%     50%     75%   97.5%  Rhat n.eff
# Y[1]      4.284   4.124  -5.646   2.400   4.630   7.313  10.745 1.411     7
# Y[2]      4.634   3.515  -4.088   2.936   5.143   7.330   9.576 1.350     9
# Y[3]      4.985   2.970  -2.635   3.400   5.833   7.307   8.693 1.277    12
# Y[4]      5.335   2.502  -1.108   3.892   6.284   7.281   8.348 1.214    21
# Y[5]      5.687   2.138   0.368   4.516   6.119   7.330   8.578 1.196   140
# Y[6]      6.041   1.927   1.407   5.035   5.972   7.383   9.120 1.257   210
# Y[7]      6.397   1.917   2.350   5.203   6.279   7.795   9.761 1.258    43
# Y[8]      6.757   2.085   3.071   5.412   6.452   8.571  10.586 1.401     8
# Y[9]      7.121   2.368   3.119   5.506   6.611   9.477  11.378 1.708     5
# Y[10]     7.489   2.700   2.810   5.950   6.765  10.296  12.110 1.975     4

# For each parameter, n.eff is a crude measure of effective sample size,
# and Rhat is the potential scale reduction factor (at convergence, Rhat=1).

# DIC info (using the rule, pD = var(deviance)/2)
# pD = 3174.9 and DIC = 12160.6
# DIC is an estimate of expected predictive error (lower deviance is better).



# Create an MCMC object from the output of the heathrow model
jags.mcmc.heathrow2 <- as.mcmc(jags.mod.fit.heathrow2)
# Graphical parameters
par(mar= c(2,4,4,2), cex=1.0)
layout(matrix(c(1,1,2,2),2,2,byrow=TRUE))
# for a traceplot, producing for groups of 100 traceplots
traceplot(jags.mcmc.heathrow2[,101:150], params = c("sigma2_3", "sigma2_4", "Y"))
traceplot(jags.mcmc.heathrow2[,251:300], params = c("sigma2_3", "sigma2_4", "Y"))
```

```
traceplot(jags.mcmc.heathrow2[,1:20], params = c("sigma2_3", "sigma2_4", "Y"))

# for summary statistics
summary(jags.mcmc.heathrow2)
# Iterations = 5001:10000
# Thinning interval = 1
# Number of chains = 2
# Sample size per chain = 5000

# 1. Empirical mean and standard deviation for each variable, plus standard error of the mean:
#              Mean       SD  Naive SE  Time-series SE
# deviance  9.000e+03 116.07895 1.1607895     37.039459
# sigma2_3  2.895e+02  33.46901 0.3346901      7.456928
# sigma2_4  4.763e-02   0.09978 0.0009978      0.008444
# Y[1]      4.284e+00   4.12365 0.0412365      1.415694
# Parameters with associated measurements
# Y[1150]  -1.084e+00   1.9156  0.019156       0.534265
# Y[1151]  -1.073e+00   1.9635  0.019635       0.578673
# Y[1152]  -9.978e-01   1.9939  0.019939       0.546461
# Y[1153]  -8.569e-01   2.0115  0.020115       0.676560
# Y[1154]  -6.522e-01   2.0220  0.020220       0.684998
# Y[1155]  -3.997e-01   2.0179  0.020179       0.629474
# Y[1156]  -1.197e-01   1.9839  0.019839       0.689137
# Y[1157]   1.669e-01   1.9128  0.019128       0.649306
# Y[1158]   4.389e-01   1.8100  0.018100       0.612609
# Y[1159]   6.707e-01   1.6987  0.016987       0.571403
# Parameters with missing data
# Y[1160]   8.393e-01   1.6079  0.016079       0.621090
# Y[1161]   9.321e-01   1.5541  0.015541       0.692370
# Y[1162]   9.492e-01   1.5509  0.015509       0.597600
# Y[1163]   8.965e-01   1.5981  0.015981       0.551568
# Y[1164]   7.857e-01   1.6898  0.016898       0.569770
# Y[1165]   6.246e-01   1.8143  0.018143       0.557858
# Y[1166]   4.202e-01   1.9576  0.019576       0.503305
# Y[1167]   1.805e-01   2.1082  0.021082       0.493607
# Y[1168]  -9.006e-02   2.2542  0.022542       0.495192
# Y[1169]  -3.831e-01   2.3947  0.023947       0.481737

# 2. Quantiles for each variable:
#              2.5%       25%       50%       75%     97.5%
# deviance 8797.64658  8.875e+03  9.043e+03 9089.13549 9175.79280
# sigma2_3  231.73188  2.586e+02  2.948e+02  314.43489  348.16491
# sigma2_4    0.01073  1.570e-02  2.637e-02    0.04942    0.20102
# Y[1]       -5.64574  2.400e+00  4.630e+00    7.31317   10.74492
# Parameters with associated measurements
# Y[1150]  -5.206e+00 -2.591e+00 -7.151e-01  3.687e-01    1.7480
# Y[1151]  -5.218e+00 -2.695e+00 -6.920e-01  4.219e-01    1.9343
# Y[1152]  -5.129e+00 -2.765e+00 -6.054e-01  5.139e-01    2.2391
# Y[1153]  -4.916e+00 -2.646e+00 -4.243e-01  6.445e-01    2.5886
# Y[1154]  -4.659e+00 -2.415e+00 -1.134e-01  8.668e-01    2.9530
# Y[1155]  -4.413e+00 -2.125e+00  1.264e-01  1.103e+00    3.2813
# Y[1156]  -4.117e+00 -1.492e+00  3.235e-01  1.283e+00    3.4846
# Y[1157]  -3.758e+00 -8.305e-01  5.450e-01  1.505e+00    3.5778
# Y[1158]  -3.323e+00 -4.647e-01  7.160e-01  1.707e+00    3.5868
# Y[1159]  -2.828e+00 -3.120e-01  9.207e-01  1.860e+00    3.5859
# Parameters with missing data
# Y[1160]  -2.416e+00 -2.077e-01  9.561e-01  1.951e+00    3.5778
```

```
# Y[1161]  -2.088e+00 -2.420e-01  8.984e-01  2.070e+00    3.7898
# Y[1162]  -1.816e+00 -3.346e-01  8.686e-01  2.068e+00    4.0318
# Y[1163]  -1.636e+00 -3.797e-01  6.855e-01  2.045e+00    4.2404
# Y[1164]  -1.567e+00 -5.177e-01  4.190e-01  1.973e+00    4.3658
# Y[1165]  -1.762e+00 -7.739e-01  1.582e-01  1.922e+00    4.5006
# Y[1166]  -2.089e+00 -1.018e+00 -2.519e-01  1.739e+00    4.5643
# Y[1167]  -2.469e+00 -1.352e+00 -7.022e-01  1.700e+00    4.4915
# Y[1168]  -2.951e+00 -1.850e+00 -1.067e+00  1.683e+00    4.3101
# Y[1169]  -3.615e+00 -2.319e+00 -1.354e+00  1.666e+00    4.0436


# plot density manually
sim.values.heathrow2 <- jags.mod.fit.heathrow2$BUGSoutput$sims.list

df.heathrow3 <- data.frame(sigma2_3 = sim.values.heathrow2$sigma2_3,
                           sigma2_4 = sim.values.heathrow2$sigma2_4)

heathrow_sigma2_3 <- ggplot(data = df.heathrow3, aes(x = sigma2_3)) +
  geom_density(colour = '#0093af', fill = '#0093af', alpha = 0.3) +
  theme(axis.title = element_blank(),
        axis.text = element_text(size = 12),
        plot.title = element_text(size = 14)) +
  geom_vline(xintercept = 0, linetype = "dashed",
             colour = "#aa0078", size = 1) +
  ggtitle('Posterior density of sigma2_3 - measurement variance')
heathrow_sigma2_3

heathrow_sigma2_4 <- ggplot(data = df.heathrow3, aes(x = sigma2_4)) +
  geom_density(colour = '#0093af', fill = '#0093af', alpha = 0.3) +
  theme(axis.title = element_blank(),
        axis.text = element_text(size = 12),
        plot.title = element_text(size = 14)) +
  geom_vline(xintercept = 0, linetype = "dashed",
             colour = "#aa0078", size = 1) +
  ggtitle('Posterior density of sigma2_4 - estimate variance')
heathrow_sigma2_4


# heathrow data for just the first quarter of 2000
heathrow3a <- heathrow3[1:91,]
tail(heathrow3a)

# Plotting the measurements and estimates for first quarter of 2000 for RW model 1
mu.heathrow2 <- jags.mod.fit.heathrow$BUGSoutput$mean$Y[1:91]
sd.heathrow2 <- jags.mod.fit.heathrow$BUGSoutput$sd$Y[1:91]

# ```{r}
df.heathrow4 <- data.frame(x2 = heathrow3a$Date, y3 = mu.heathrow2, y4 = heathrow3a$Heathrow,
                           lower.heathrow2 = mu.heathrow2 - 1.96*sd.heathrow2,
                           upper.heathrow2 = mu.heathrow2 + 1.96*sd.heathrow2)

ggplot(data = df.heathrow4) +
  geom_point(aes(x = x2, y = y3), colour = 'grey31', size = 2) +
  geom_point(aes(x = x2, y = y4), colour = 'blue', size = 2) +
  geom_line(aes(x = x2, y = y3), colour = '#0093af', size = 1) +
  geom_line(aes(x = x2, y = lower.heathrow2), linetype = "dashed", colour = "#aa0078", size = 1) +
  geom_line(aes(x = x2, y = upper.heathrow2), linetype = "dashed", colour = "#aa0078", size = 1) +
```

```
  xlab('Date') + ylab('Y (estimates) and Heathrow (measurements) of PM10 at Heathrow') +
  ggtitle('PM10 measurements and estimates at Heathrow for first quarter of 2000 with RW model 1') +
  theme(axis.title = element_text(size = 14),
        axis.text = element_text(size = 12),
        plot.title = element_text(size = 14))


# Plotting the measurements and estimates for first quarter of 2000 for RW model 2
mu.heathrow3 <- jags.mod.fit.heathrow2$BUGSoutput$mean$Y[1:91]
sd.heathrow3 <- jags.mod.fit.heathrow2$BUGSoutput$sd$Y[1:91]

# ```{r}
df.heathrow5 <- data.frame(x3 = heathrow3a$Date, y5 = mu.heathrow3, y6 = heathrow3a$Heathrow,
                           lower.heathrow3 = mu.heathrow3 - 1.96*sd.heathrow3,
                           upper.heathrow3 = mu.heathrow3 + 1.96*sd.heathrow3)

ggplot(data = df.heathrow5) +
  geom_point(aes(x = x3, y = y5), colour = 'grey31', size = 2) +
  geom_point(aes(x = x3, y = y6), colour = 'blue', size = 2) +
  geom_line(aes(x = x3, y = y5), colour = '#0093af', size = 1) +
  geom_line(aes(x = x3, y = lower.heathrow3), linetype = "dashed", colour = "#aa0078", size = 1) +
  geom_line(aes(x = x3, y = upper.heathrow3), linetype = "dashed", colour = "#aa0078", size = 1) +
  xlab('Date') + ylab('Y (estimates) and Heathrow (measurements) of PM10 at Heathrow') +
  ggtitle('PM10 measurements and estimates at Heathrow for first quarter of 2000 with RW model 2') +
  theme(axis.title = element_text(size = 14),
        axis.text = element_text(size = 12),
        plot.title = element_text(size = 14))


# Produce an MCMC trace of sigma2_3 output
MCMCtrace(jags.mcmc.heathrow2,
          params = c('sigma2_3'),
          type = 'density',
          ind = TRUE,
          ISB = FALSE,
          pdf = FALSE,
          col_den = c("#aa0078", "#0093af"))

# Produce an MCMC trace of sigma2_4 output
MCMCtrace(jags.mcmc.heathrow2,
          params = c('sigma2_4'),
          type = 'density',
          ind = TRUE,
          ISB = FALSE,
          pdf = FALSE,
          col_den = c("#aa0078", "#0093af"))

# Produce an MCMC trace of Y output
MCMCtrace(jags.mcmc.heathrow2,
          params = c('Y'),
          type = 'density',
          ind = TRUE,
          ISB = FALSE,
          pdf = FALSE,
          col_den = c("#aa0078", "#0093af"))
```

```
gelman.heathrow2 <- gelman.diag(jags.mcmc.heathrow2)
gelman.heathrow2
# Potential scale reduction factors:
#          Point est. Upper C.I.
# deviance       2.09       5.57
# sigma2_3       1.81       3.61
# sigma2_4       1.02       1.08
# Y[1]           1.75       3.09
# Parameters with associated measurements
# Y[1090]        1.31       2.09
# Y[1091]        1.58       2.83
# Y[1092]        1.96       3.70
# Y[1093]        2.39       4.69
# Y[1094]        2.74       5.79
# Y[1095]        2.90       6.92
# Y[1096]        2.84       7.57
# Y[1097]        2.65       7.61
# Y[1098]        2.41       7.41
# Y[1099]        2.11       6.59
# Parameters with missing data
# Y[1100]        1.71       4.68
# Y[1101]        1.32       2.48
# Y[1102]        1.13       1.16
# Y[1103]        1.18       1.66
# Y[1104]        1.63       3.02
# Y[1105]        2.24       4.56
# Y[1106]        2.79       5.91
# Y[1107]        3.26       6.95
# Y[1108]        3.63       7.74
# Y[1109]        3.92       8.35

# Multivariate psrf
# 107

# Create a data.frame from the gelman.diag output to put into a kable table
rhat.heathrow2 <- as.data.frame((gelman.heathrow2)$psrf)
kbl(rhat.heathrow2) %>% kable_styling()

# Create a gelman plot of all the parameters and their densities
gelman.plot(jags.mcmc.heathrow2)



# With Random Walk 1 modelling, the estimates are based on the original measurements, so there is
# something like one degree of "freedom" or separation. With random walk 2 modelling, estimates are
# based on on previous estimates, that themselves are based on the measurements. This provides something
# like two degrees of "freedom" or separation, so the chains do not converge as there is less to
# constrain them. Analogous scenarios to this include:
# 1) a single pendulum swinging (i.e. RW1) and a pendulum attached to another pendulum with both swinging
# in a "choatic manner" (i.e. RW2); or
# 2) atoms in a solid state, who can vibrate within a fixed amount of space (and converge) around a point
# but are constrained by neighbouring atoms (i.e. RW1) and atoms in a liquid state, who can vibrate and move
# relative to each other in a "choatic manner", so don't stay in a fixed space (and converge) to point (i.e. RW2).
```

8. [8 marks] Use both of your models to predict the measurements of PM10 at Heathrow for the first week of 2004.

```
## Reading in London pollution data
london_pollution <- read.csv("London_Pollution.csv")
london_pollution$Date <- as.Date(london_pollution$Date, format = "%d/%m/%Y")
head(london_pollution)
tail(london_pollution)
london_pollution[350:400,]
london_pollution[50:100,]

## The date and heathrow air pollution data are in the 2nd and 3rd columns, respectively,
## and there are 1827 measurements from 1st January 2000 to 31st December 2004:
heathrow <- dplyr::select(london_pollution, Date, Heathrow)
head(heathrow)
tail(heathrow)
count(heathrow)
sapply(heathrow, class)

# Creating new Day, Month and Year columns from Date column for Heathrow data
heathrow <- heathrow %>% separate(Date, into = c('Year', 'Month', 'Day'))
heathrow2 <- cbind(heathrow, Date = london_pollution$Date)
head(heathrow2)
# arrange columns with Date, Day, Month, Year, Heathrow air pollution
heathrow2 <- heathrow2 %>% dplyr::select(Date, Year, Month, Day, Heathrow)
head(heathrow2)
tail(heathrow2)

# heathrow data for just 2000-2003 and first week in 2004
heathrow4 <- heathrow2 %>% slice(1:1468)
tail(heathrow4)
heathrow4[1001:1100,]
heathrow4[1101:1200,]
heathrow4[1201:1300,]
heathrow4[1450:1470,]
# Set seed for reproducibility
set.seed(234)
# Set N to be the length of the data
N2 <- 1468
# Mean of Heathrow data
mean.heathrow2 <- mean(heathrow4$Heathrow, na.rm = TRUE)
mean.heathrow2
# Extract Heathrow column data
Heathrow2 <- heathrow4$Heathrow
tail(Heathrow2)
# Convert last 7 rows to NA values (i.e. for 1st week of 2004)
Heathrow2[1462:1468] <- NA
Heathrow2[1450:1468]

# RW1 model

# List the data to be used
jags.data.heathrow3 <- list("Heathrow2", "N2")

# Model
jags.mod.heathrow3 <- function(){
  Y2[1] ~ dnorm(0, 0.001)
  for(t in 2:N2){
    Heathrow2[t] ~ dnorm(Y2[t], tau5)            # normal likelihood of heathrow data
```

```
   Y2[t] ~ dnorm(Y2[t-1], tau6)              # normal prediction model
  }
  # priors on measurement error and white noise variances
  tau5 ~ dgamma(0.001, 0.001)                 # gamma measurement error precision model
  sigma2_5 <- 1/tau5                        # measurement error variance
  tau6 ~ dgamma(0.001, 0.001)                 # gamma estimate error precision model
  sigma2_6 <- 1/tau6                        # estimate error variance
}

# Specify initial values
inits.heathrow5 <- list("Y2" = c(rep.int(NA,1461), rep.int(mean.heathrow2,7)), "is.na(Heathrow)" = mean.heathrow2)
inits.heathrow6 <- list("Y2" = c(rep.int(NA,1461), rep.int(21,7)), "is.na(Heathrow)" = mean.heathrow2)
jags.inits.heathrow3 <- list(inits.heathrow5, inits.heathrow6)

# Monitor the parameters to be used for the prediction
jags.param.heathrow3 <- c("sigma2_5","sigma2_6","Y2")

# Fitting the new model
jags.mod.fit.heathrow3 <- jags(data = jags.data.heathrow3, inits = jags.inits.heathrow3,
                      parameters.to.save = jags.param.heathrow3, n.chains = 2,
                      n.iter = 10000, n.burnin = 5000, n.thin = 1,
                      model.file = jags.mod.heathrow3)

# Get point and interval estimates
print(jags.mod.fit.heathrow3)
# Inference for Bugs model at
"/var/folders/pc/hksslngn6_56y6dc8464zthm0000gn/T//RtmpuRFXJq/modelb56730bf50.txt", fit using jags,
# 2 chains, each with 10000 iterations (first 5000 discarded)
# n.sims = 10000 iterations saved
# mu.vect sd.vect    2.5%     25%     50%     75%   97.5%  Rhat n.eff
# Y2[1]     10.583   6.247  -1.804   6.479  10.594  14.809  22.824 1.003   830
# Y2[2]     10.924   4.017   2.966   8.248  10.903  13.600  18.768 1.003   830
# Y2[3]     10.887   3.573   3.935   8.467  10.908  13.275  17.937 1.002  1500
# Y2[4]     13.005   3.431   6.250  10.732  12.994  15.311  19.697 1.001  3600
# Y2[5]     16.983   3.441  10.197  14.737  16.997  19.302  23.720 1.001  5700
# Y2[6]     18.564   3.393  11.758  16.320  18.593  20.881  25.017 1.001  3100
# Y2[7]     20.193   3.436  13.492  17.946  20.193  22.443  26.978 1.001 10000
# Y2[8]     21.188   3.480  14.397  18.826  21.141  23.540  27.989 1.001  6200
# Y2[9]     26.813   3.589  19.572  24.531  26.852  29.179  33.700 1.002  7500
# Y2[10]    29.827   3.999  21.609  27.316  29.903  32.506  37.383 1.002 10000
# For each parameter, n.eff is a crude measure of effective sample size,
# and Rhat is the potential scale reduction factor (at convergence, Rhat=1).

# DIC info (using the rule, pD = var(deviance)/2)
# pD = 21535.7 and DIC = 28088.6
# DIC is an estimate of expected predictive error (lower deviance is better).


# plot density manually
sim.values.heathrow3 <- jags.mod.fit.heathrow3$BUGSoutput$sims.list

df.heathrow6 <- data.frame(sigma2_5 = sim.values.heathrow3$sigma2_5,
                    sigma2_6 = sim.values.heathrow3$sigma2_6)

heathrow_sigma2_5 <- ggplot(data = df.heathrow6, aes(x = sigma2_5)) +
  geom_density(colour = '#0093af', fill = '#0093af', alpha = 0.3) +
  theme(axis.title = element_blank(),
```

```
      axis.text = element_text(size = 12),
      plot.title = element_text(size = 14)) +
  geom_vline(xintercept = 0, linetype = "dashed",
          colour = "#aa0078", size = 1) +
  ggtitle('Posterior density of sigma2_5 - measurement variance')
heathrow_sigma2_5

heathrow_sigma2_6 <- ggplot(data = df.heathrow6, aes(x = sigma2_6)) +
  geom_density(colour = '#0093af', fill = '#0093af', alpha = 0.3) +
  theme(axis.title = element_blank(),
      axis.text = element_text(size = 12),
      plot.title = element_text(size = 14)) +
  geom_vline(xintercept = 0, linetype = "dashed",
          colour = "#aa0078", size = 1) +
  ggtitle('Posterior density of sigma2_6 - estimate variance')
heathrow_sigma2_6


# Create an MCMC object from the output of the heathrow model
jags.mcmc.heathrow3 <- as.mcmc(jags.mod.fit.heathrow3)
# Graphical parameters
par(mar = c(2,4,4,2), cex = 1.0)
layout(matrix(c(1,1,2,2),2,2,byrow=TRUE))
# for a traceplot, producing for groups of traceplots
traceplot(jags.mcmc.heathrow3[,510:530], params = c("sigma2_5", "sigma2_6", "Y2"))
traceplot(jags.mcmc.heathrow3[,1:4], params = c("sigma2_5", "sigma2_6", "Y2"))
traceplot(jags.mcmc.heathrow3[,101:150], params = c("sigma2_5", "sigma2_6", "Y2"))
traceplot(jags.mcmc.heathrow3[,251:300], params = c("sigma2_5", "sigma2_6", "Y2"))

# for summary statistics
summary(jags.mcmc.heathrow3)
# Iterations = 5001:10000
# Thinning interval = 1
# Number of chains = 2
# Sample size per chain = 5000

# 1. Empirical mean and standard deviation for each variable, plus standard error of the mean:
#           Mean     SD  Naive SE  Time-series SE
# deviance 6552.928 207.744  2.07744       18.10619
# sigma2_5   27.904   8.975  0.08975       0.74029
# sigma2_6   24.520   5.193  0.05193       0.49561
# Y2[1]      10.583   6.247  0.06247       0.10271
# Parameters with associated measurements
# Y2[1150]   12.160  20.429  0.20429       3.85553
# Y2[1151]   12.107  20.429  0.20429       3.68885
# Y2[1152]   11.887  20.411  0.20411       3.76813
# Y2[1153]   11.687  20.403  0.20403       3.79963
# Y2[1154]   11.599  20.342  0.20342       3.80123
# Y2[1155]   11.575  20.275  0.20275       3.47538
# Y2[1156]   11.539  20.176  0.20176       3.52933
# Y2[1157]   11.540  20.150  0.20150       3.76458
# Y2[1158]   11.472  20.146  0.20146       3.75236
# Y2[1159]   11.369  20.152  0.20152       3.64520
# Parameters with missing data
# Y2[1160]   11.276  20.147  0.20147       3.79544
# Y2[1161]   11.166  20.069  0.20069       3.69384
# Y2[1162]   10.893  19.926  0.19926       3.38308
```

```
# Y2[1163]   10.668  19.814  0.19814       3.18778
# Y2[1164]   10.444  19.754  0.19754       3.30458
# Y2[1165]   10.227  19.592  0.19592       3.26336
# Y2[1166]    9.978  19.469  0.19469       3.23755
# Y2[1167]    9.747  19.321  0.19321       3.42372
# Y2[1168]    9.540  19.156  0.19156       3.46290
# Y2[1169]    9.411  19.011  0.19011       3.53783

# 2. Quantiles for each variable:
#              2.5%       25%      50%      75%    97.5%
# deviance 6222.24393  6.441e+03 6549.115 6640.155 6872.52
# sigma2_5   19.87302  2.434e+01   27.076   29.676   37.07
# sigma2_6   13.86291  2.156e+01   24.428   27.710   34.46
# Y2[1]     -1.80388  6.479e+00   10.594   14.809   22.82
# Parameters with associated measurements
# Y2[1150]  -28.70769 -1.017e+00   11.971   26.721   51.58
# Y2[1151]  -28.08983 -1.412e+00   12.135   26.991   50.19
# Y2[1152]  -27.75876 -2.212e+00   11.980   26.999   49.03
# Y2[1153]  -28.22241 -2.914e+00   11.984   27.497   48.66
# Y2[1154]  -27.89977 -3.409e+00   12.040   27.564   47.45
# Y2[1155]  -27.86566 -3.478e+00   12.242   27.626   46.84
# Y2[1156]  -27.64406 -3.347e+00   12.372   27.593   45.75
# Y2[1157]  -28.01363 -3.685e+00   12.384   27.691   45.74
# Y2[1158]  -27.33103 -3.811e+00   12.231   27.586   45.41
# Y2[1159]  -27.10663 -3.843e+00   12.205   27.752   45.18
# Parameters with missing data
# Y2[1160]  -27.48751 -3.860e+00   12.319   27.775   44.56
# Y2[1161]  -27.70400 -3.872e+00   12.389   27.296   44.28
# Y2[1162]  -27.62243 -4.145e+00   12.143   26.752   44.22
# Y2[1163]  -27.69856 -4.354e+00   12.031   26.438   43.99
# Y2[1164]  -27.66786 -4.578e+00   11.847   26.158   44.27
# Y2[1165]  -27.29767 -4.750e+00   11.109   25.696   44.60
# Y2[1166]  -26.51952 -5.123e+00   10.639   25.286   44.75
# Y2[1167]  -26.35066 -4.976e+00    9.986   24.903   45.08
# Y2[1168]  -25.91095 -4.864e+00    9.809   24.351   45.09
# Y2[1169]  -25.64899 -5.047e+00    9.639   23.807   45.40


# Produce an MCMC trace of sigma2_5 output
MCMCtrace(jags.mcmc.heathrow3,
        params = c('sigma2_5'),
        type = 'density',
        ind = TRUE,
        ISB = FALSE,
        pdf = FALSE,
        col_den = c("#aa0078", "#0093af"))

# Produce an MCMC trace of sigma2_6 output
MCMCtrace(jags.mcmc.heathrow3,
        params = c('sigma2_6'),
        type = 'density',
        ind = TRUE,
        ISB = FALSE,
        pdf = FALSE,
        col_den = c("#aa0078", "#0093af"))

# Produce an MCMC trace of Y2 output
```

```
MCMCtrace(jags.mcmc.heathrow3,
        params = c('Y2'),
        type = 'density',
        ind = TRUE,
        ISB = FALSE,
        pdf = FALSE,
        col_den = c("#aa0078", "#0093af"))

gelman.heathrow3 <- gelman.diag(jags.mcmc.heathrow3)
gelman.heathrow3
# Potential scale reduction factors:
#          Point est. Upper C.I.
# deviance       1.01       1.03
# sigma2_5       1.00       1.01
# sigma2_6       1.01       1.02
# Y2[1]          1.00       1.00
# Parameters with associated measurements
# Y2[1090]       1.00       1.00
# Y2[1091]       1.00       1.00
# Y2[1092]       1.00       1.00
# Y2[1093]       1.00       1.00
# Y2[1094]       1.00       1.00
# Y2[1095]       1.00       1.00
# Y2[1096]       1.00       1.00
# Y2[1097]       1.00       1.00
# Y2[1098]       1.01       1.01
# Y2[1099]       1.03       1.14
# Parameters with missing data
# Y2[1100]       1.08       1.32
# Y2[1101]       1.13       1.47
# Y2[1102]       1.17       1.59
# Y2[1103]       1.21       1.72
# Y2[1104]       1.26       1.84
# Y2[1105]       1.30       1.94
# Y2[1106]       1.33       2.05
# Y2[1107]       1.37       2.14
# Y2[1108]       1.41       2.26
# Y2[1109]       1.44       2.36

# Multivariate psrf
# 2.94

# Create a data.frame from the gelman.diag output to put into a kable table
rhat.heathrow3 <- as.data.frame((gelman.heathrow3)$psrf)
kbl(rhat.heathrow3) %>% kable_styling()

# Create a gelman plot of all the parameters and their densities
gelman.plot(jags.mcmc.heathrow3)



# Measurements for the first week of 2004
# [1] 11.6 22.1 23.1 23.6 16.8 15.9 15.4
measure.heathrow3 <- Heathrow2[1462:1468]
measure.heathrow3

# Estimates for the 1st week of 2004 for RW1, which converges around the same number (~17.5)
# [1] 17.42978 17.49028 17.52667 17.49649 17.50711 17.50969 17.49314
```

```
est.heathrow3 <- jags.mod.fit.heathrow3$BUGSoutput$mean$Y[1462:1468]
est.heathrow3



# RW2 model

# List the data to be used
jags.data.heathrow4 <- list("Heathrow2", "N2")

# Model 2
jags.mod.heathrow4 <- function(){
  Y2[1] ~ dnorm(0, 0.001)
  Y2[2] ~ dnorm(0, 0.001)
  for(t in 3:N2){
    Heathrow2[t] ~ dnorm(Y2[t], tau7)            # normal likelihood of heathrow data
    Y2[t] ~ dnorm((2*Y2[t-1] - Y2[t-2]), tau8)      # normal prediction model
  }
  # priors on measurement error and white noise variances
  tau7 ~ dgamma(0.001, 0.001)                 # gamma measurement error precision model
  sigma2_7 <- 1/tau7                       # measurement error variance
  tau8 ~ dgamma(0.001, 0.001)                 # gamma estimate error precision model
  sigma2_8 <- 1/tau8                       # estimate error variance
}

# Specify initial values
inits.heathrow7 <- list("Y2" = c(rep.int(NA,1461), rep.int(mean.heathrow2,7)), "is.na(Heathrow)" = mean.heathrow2)
inits.heathrow8 <- list("Y2" = c(rep.int(NA,1461), rep.int(21,7)), "is.na(Heathrow)" = mean.heathrow2)
jags.inits.heathrow4 <- list(inits.heathrow7, inits.heathrow8)

# Monitor the parameters to be used for the prediction
jags.param.heathrow4 <- c("sigma2_7","sigma2_8","Y2")

# Fitting the new model
jags.mod.fit.heathrow4 <- jags(data = jags.data.heathrow4, inits = jags.inits.heathrow4,
                      parameters.to.save = jags.param.heathrow4, n.chains = 2,
                      n.iter = 10000, n.burnin = 5000, n.thin = 1,
                      model.file = jags.mod.heathrow4)

# Get point and interval estimates
print(jags.mod.fit.heathrow4)
# Inference for Bugs model at
"/var/folders/pc/hksslngn6_56y6dc8464zthm0000gn/T//RtmpuRFXJq/modelb565e72a2ec.txt", fit using jags,
# 2 chains, each with 10000 iterations (first 5000 discarded)
# n.sims = 10000 iterations saved
#         mu.vect sd.vect    2.5%     25%     50%     75%   97.5%  Rhat n.eff
# Y2[1]     7.445   4.366  -2.132   4.114   7.961  11.049  14.123 2.755     3
# Y2[2]     7.401   3.900  -0.961   4.227   7.877  10.676  13.279 3.021     3
# Y2[3]     7.358   3.482   0.151   4.296   7.875  10.330  12.537 3.325     3
# Y2[4]     7.317   3.122   1.228   4.465   7.906  10.075  11.846 2.650     3
# Y2[5]     7.282   2.816   2.170   4.626   7.686   9.823  11.254 3.272     3
# Y2[6]     7.260   2.558   3.018   4.732   7.557   9.581  10.751 3.559     3
# Y2[7]     7.264   2.343   3.214   4.924   7.516   9.416  10.346 3.584     3
# Y2[8]     7.302   2.167   3.420   5.319   7.676   9.278  10.234 3.334     3
# Y2[9]     7.385   2.017   3.666   5.639   7.843   9.158  10.171 2.874     3
# Y2[10]    7.519   1.890   3.715   5.985   8.113   9.036  10.181 2.391     3
```

```
# For each parameter, n.eff is a crude measure of effective sample size,
# and Rhat is the potential scale reduction factor (at convergence, Rhat=1).

# DIC info (using the rule, pD = var(deviance)/2)
# pD = 1299.8 and DIC = 10308.8
# DIC is an estimate of expected predictive error (lower deviance is better).


# plot density manually
sim.values.heathrow4 <- jags.mod.fit.heathrow4$BUGSoutput$sims.list

df.heathrow8 <- data.frame(sigma2_7 = sim.values.heathrow4$sigma2_7,
                           sigma2_8 = sim.values.heathrow4$sigma2_8)

heathrow_sigma2_7 <- ggplot(data = df.heathrow8, aes(x = sigma2_7)) +
  geom_density(colour = '#0093af', fill = '#0093af', alpha = 0.3) +
  theme(axis.title = element_blank(),
        axis.text = element_text(size = 12),
        plot.title = element_text(size = 14)) +
  geom_vline(xintercept = 0, linetype = "dashed",
             colour = "#aa0078", size = 1) +
  ggtitle('Posterior density of sigma2_7 - measurement variance')
heathrow_sigma2_7

heathrow_sigma2_8 <- ggplot(data = df.heathrow8, aes(x = sigma2_8)) +
  geom_density(colour = '#0093af', fill = '#0093af', alpha = 0.3) +
  theme(axis.title = element_blank(),
        axis.text = element_text(size = 12),
        plot.title = element_text(size = 14)) +
  geom_vline(xintercept = 0, linetype = "dashed",
             colour = "#aa0078", size = 1) +
  ggtitle('Posterior density of sigma2_8 - estimate variance')
heathrow_sigma2_8


# Create an MCMC object from the output of the heathrow model
jags.mcmc.heathrow4 <- as.mcmc(jags.mod.fit.heathrow4)
# Graphical parameters
par(mar = c(2,4,4,2), cex = 1.0)
layout(matrix(c(1,1,2,2),2,2,byrow=TRUE))
# for a traceplot, producing for groups of traceplots
traceplot(jags.mcmc.heathrow4[,510:530], params = c("sigma2_7", "sigma2_8", "Y2"))
traceplot(jags.mcmc.heathrow4[,1:4], params = c("sigma2_7", "sigma2_8", "Y2"))
traceplot(jags.mcmc.heathrow4[,101:150], params = c("sigma2_7", "sigma2_8", "Y2"))
traceplot(jags.mcmc.heathrow4[,251:300], params = c("sigma2_7", "sigma2_8", "Y2"))

# for summary statistics
summary(jags.mcmc.heathrow4)
# Iterations = 5001:10000
# Thinning interval = 1
# Number of chains = 2
# Sample size per chain = 5000

# 1. Empirical mean and standard deviation for each variable, plus standard error of the mean:
#              Mean      SD  Naive SE  Time-series SE
# deviance  8.953e+03 105.1115 1.051115      45.689417
# sigma2_7  2.616e+02  28.1689 0.281689      10.874633
```

```
# sigma2_8  5.491e-02   0.1101 0.001101      0.009114
# Y2[1]     7.445e+00   4.3664 0.043664      0.577472
# Parameters with associated measurements
# Y2[1150] -3.499e+00   1.3288 0.013288      0.411425
# Y2[1151] -3.301e+00   1.4010 0.014010      0.442363
# Y2[1152] -3.053e+00   1.5214 0.015214      0.507581
# Y2[1153] -2.766e+00   1.6747 0.016747      0.598440
# Y2[1154] -2.446e+00   1.8292 0.018292      0.655132
# Y2[1155] -2.106e+00   1.9477 0.019477      0.714047
# Y2[1156] -1.764e+00   2.0036 0.020036      0.712516
# Y2[1157] -1.434e+00   1.9814 0.019814      0.722113
# Y2[1158] -1.125e+00   1.8848 0.018848      0.674243
# Parameters with missing data
# Y2[1160] -6.047e-01   1.5196 0.015196      0.486243
# Y2[1161] -3.945e-01   1.3145 0.013145      0.403056
# Y2[1162] -2.094e-01   1.1806 0.011806      0.311593
# Y2[1163] -5.082e-02   1.1964 0.011964      0.272011
# Y2[1164]  7.223e-02   1.3829 0.013829      0.204829
# Y2[1165]  1.482e-01   1.6882 0.016882      0.181720
# Y2[1166]  1.726e-01   2.0524 0.020524      0.184789
# Y2[1167]  1.488e-01   2.4333 0.024333      0.210708
# Y2[1168]  8.949e-02   2.7942 0.027942      0.232273
# Y2[1169] -2.730e-03   3.1144 0.031144      0.255136

# 2. Quantiles for each variable:
#              2.5%       25%       50%       75%     97.5%
# deviance 8740.49943  8.905e+03  8.952e+03  9.019e+03 9172.85114
# sigma2_7  208.50112  2.443e+02  2.615e+02  2.785e+02  321.99795
# sigma2_8    0.01286  1.933e-02  3.299e-02  5.107e-02    0.24291
# Y2[1]      -2.13175  4.114e+00  7.961e+00  1.105e+01   14.12321
# Parameters with associated measurements
# Y2[1150]   -5.41410 -4.503e+00 -3.692e+00 -2.877e+00   -0.03230
# Y2[1151]   -5.16137 -4.333e+00 -3.691e+00 -2.479e+00    0.27066
# Y2[1152]   -4.98581 -4.219e+00 -3.598e+00 -2.041e+00    0.46165
# Y2[1153]   -4.76440 -4.122e+00 -3.494e+00 -1.459e+00    0.98246
# Y2[1154]   -4.57083 -3.946e+00 -3.357e+00 -8.472e-01    1.45011
# Y2[1155]   -4.37975 -3.708e+00 -3.141e+00 -3.528e-01    1.97247
# Y2[1156]   -4.13188 -3.423e+00 -2.819e+00  6.737e-02    2.35852
# Y2[1157]   -3.86170 -3.058e+00 -2.433e+00  3.950e-01    2.52482
# Y2[1158]   -3.59914 -2.656e+00 -1.957e+00  6.339e-01    2.60827
# Y2[1159]   -3.45096 -2.198e+00 -1.439e+00  7.575e-01    2.41551
# Parameters with missing data
# Y2[1160]   -3.41134 -1.676e+00 -9.045e-01  7.714e-01    2.07698
# Y2[1161]   -3.37832 -1.139e+00 -4.028e-01  6.825e-01    1.75404
# Y2[1162]   -3.37457 -6.494e-01 -4.008e-03  5.457e-01    1.59340
# Y2[1163]   -3.34878 -4.425e-01  1.823e-01  6.704e-01    1.77192
# Y2[1164]   -3.39114 -8.407e-01  3.638e-01  1.097e+00    2.20911
# Y2[1165]   -3.51352 -1.179e+00  5.771e-01  1.585e+00    2.67524
# Y2[1166]   -3.67365 -1.535e+00  5.158e-01  1.995e+00    3.16527
# Y2[1167]   -3.91195 -1.983e+00  3.748e-01  2.356e+00    3.53502
# Y2[1168]   -4.28557 -2.495e+00  2.174e-01  2.654e+00    3.89170
# Y2[1169]   -4.78018 -2.979e+00  5.728e-02  2.903e+00    4.23912


# Produce an MCMC trace of sigma2_7 output
MCMCtrace(jags.mcmc.heathrow4,
        params = c('sigma2_7'),
```

```
        type = 'density',
        ind = TRUE,
        ISB = FALSE,
        pdf = FALSE,
        col_den = c("#aa0078", "#0093af"))

# Produce an MCMC trace of sigma2_8 output
MCMCtrace(jags.mcmc.heathrow4,
        params = c('sigma2_8'),
        type = 'density',
        ind = TRUE,
        ISB = FALSE,
        pdf = FALSE,
        col_den = c("#aa0078", "#0093af"))

# Produce an MCMC trace of Y2 output
MCMCtrace(jags.mcmc.heathrow4,
        params = c('Y2'),
        type = 'density',
        ind = TRUE,
        ISB = FALSE,
        pdf = FALSE,
        col_den = c("#aa0078", "#0093af"))

gelman.heathrow4 <- gelman.diag(jags.mcmc.heathrow4)
gelman.heathrow4
# Potential scale reduction factors:
#           Point est. Upper C.I.
# deviance       1.05       1.06
# sigma2_7       1.02       1.02
# sigma2_8       1.00       1.00
# Y2[1]          4.14       9.09
# Parameters with associated measurements
# Y2[1090]       1.29       2.14
# Y2[1091]       1.14       1.35
# Y2[1092]       1.11       1.16
# Y2[1093]       1.27       2.02
# Y2[1094]       1.86       3.69
# Y2[1095]       2.72       5.68
# Y2[1096]       3.65       7.73
# Y2[1097]       4.54       9.84
# Y2[1098]       5.39      12.12
# Y2[1099]       5.90      13.68
# Parameters with missing data
# Y2[1100]       5.93      14.22
# Y2[1101]       5.77      14.39
# Y2[1102]       5.58      14.51
# Y2[1103]       5.44      14.73
# Y2[1104]       5.34      14.87
# Y2[1105]       5.31      14.73
# Y2[1106]       5.29      14.11
# Y2[1107]       5.24      13.05
# Y2[1108]       5.16      11.90
# Y2[1109]       5.08      11.14

# Multivariate psrf
# 116
```

```
# Create a data.frame from the gelman.diag output to put into a kable table
rhat.heathrow4 <- as.data.frame((gelman.heathrow4)$psrf)
kbl(rhat.heathrow4) %>% kable_styling()

# Create a gelman plot of all the parameters and their densities
gelman.plot(jags.mcmc.heathrow4)


# Measurements for the first week of 2004
# [1] 11.6 22.1 23.1 23.6 16.8 15.9 15.4
measure.heathrow4 <- Heathrow2[1462:1468]
measure.heathrow4

# Estimates for the 1st week of 2004 for RW2, which do not converge on a single number
# [1] 1.4998122 1.4064798 1.3143710 1.2232489 1.1331479 1.0420770 0.9529367
est.heathrow4 <- jags.mod.fit.heathrow4$BUGSoutput$mean$Y[1462:1468]
est.heathrow4



## 9. [6 marks] For both models, plot the predicted values of PM10 for the first week of 2004, along with the actual
## measurements, against time. By calculating appropriate measures of comparison, comment on how good you
think the
## models are at forecasting. Hint: you may want to re-run the model with an
## sqrt(sum((Yhatt − Y)^2 / n))
## extra line to calculate the root mean squared prediction error, noting that this value will also have a posterior
## distribution as it is a function of the predicted values (that are treated as unknown paramaters that need to be
estimated).


# heathrow data for just the last quarter of 2003 and first week of 2004
heathrow4a <- heathrow4[1370:1468,]
tail(heathrow4a)

# Plotting the measurements and estimates for last quarter of 2003 and first week of 2004 for RW model 1
mu.heathrow4 <- jags.mod.fit.heathrow3$BUGSoutput$mean$Y[1370:1468]
sd.heathrow4 <- jags.mod.fit.heathrow3$BUGSoutput$sd$Y[1370:1468]

# ```{r}
df.heathrow7 <- data.frame(x4 = heathrow4a$Date, y7 = mu.heathrow4, y8 = heathrow4a$Heathrow,
                           lower.heathrow4 = mu.heathrow4 - 1.96*sd.heathrow4,
                           upper.heathrow4 = mu.heathrow4 + 1.96*sd.heathrow4)

ggplot(data = df.heathrow7) +
  geom_point(aes(x = x4, y = y7), colour = 'grey31', size = 2) +
  geom_point(aes(x = x4, y = y8), colour = 'blue', size = 2) +
  geom_line(aes(x = x4, y = y7), colour = '#0093af', size = 1) +
  geom_line(aes(x = x4, y = lower.heathrow4), linetype = "dashed", colour = "#aa0078", size = 1) +
  geom_line(aes(x = x4, y = upper.heathrow4), linetype = "dashed", colour = "#aa0078", size = 1) +
  xlab('Date') + ylab('Y (estimates) and Heathrow (measurements) of PM10 at Heathrow') +
  ggtitle('PM10 measurements and estimates at Heathrow for last quarter of 2003 and first week of 2004 with RW
model 1') +
  theme(axis.title = element_text(size = 14),
        axis.text = element_text(size = 12),
        plot.title = element_text(size = 14))
```

```r
# Plotting the measurements and estimates for last quarter of 2003 and first week of 2004 for RW model 1
mu.heathrow5 <- jags.mod.fit.heathrow4$BUGSoutput$mean$Y[1370:1468]
sd.heathrow5 <- jags.mod.fit.heathrow4$BUGSoutput$sd$Y[1370:1468]

# ```{r}
df.heathrow9 <- data.frame(x5 = heathrow4a$Date, y9 = mu.heathrow5, y10 = heathrow4a$Heathrow,
                           lower.heathrow5 = mu.heathrow5 - 1.96*sd.heathrow5,
                           upper.heathrow5 = mu.heathrow5 + 1.96*sd.heathrow5)

ggplot(data = df.heathrow9) +
  geom_point(aes(x = x5, y = y9), colour = 'grey31', size = 2) +
  geom_point(aes(x = x5, y = y10), colour = 'blue', size = 2) +
  geom_line(aes(x = x5, y = y9), colour = '#0093af', size = 1) +
  geom_line(aes(x = x5, y = lower.heathrow5), linetype = "dashed", colour = "#aa0078", size = 1) +
  geom_line(aes(x = x5, y = upper.heathrow5), linetype = "dashed", colour = "#aa0078", size = 1) +
  xlab('Date') + ylab('Y (estimates) and Heathrow (measurements) of PM10 at Heathrow') +
  ggtitle('PM10 measurements and estimates at Heathrow for last quarter of 2003 and first week of 2004 with RW
model 2') +
  theme(axis.title = element_text(size = 14),
        axis.text = element_text(size = 12),
        plot.title = element_text(size = 14))



# Y vs mean.Y

# heathrow data for just 2000-2003 and first week in 2004 with root mean square function
heathrow4 <- heathrow2 %>% slice(1:1468)
tail(heathrow4)
heathrow4[1001:1100,]
heathrow4[1101:1200,]
heathrow4[1201:1300,]
heathrow4[1450:1470,]
# Set seed for reproducibility
set.seed(234)
# Set N to be the length of the data
N2 <- 1468
# Mean of Heathrow data
mean.heathrow2 <- mean(heathrow4$Heathrow, na.rm = TRUE)
mean.heathrow2
# Extract Heathrow column data
Heathrow2 <- heathrow4$Heathrow
tail(Heathrow2)

# Mean of Y estimate data for RW1 (= 19.58195)
mean.Y <- mean(jags.mod.fit.heathrow3$BUGSoutput$mean$Y)
mean.Y


# RW1 model with root mean square

# List the data to be used
jags.data.heathrow5 <- list("Heathrow2", "N2")
```

```
# Model
jags.mod.heathrow5 <- function(){
  Y2[1] ~ dnorm(0, 0.001)
  mean.Y <- 19.58195
  for(t in 2:N2){
    Heathrow2[t] ~ dnorm(Y2[t], tau9)              # normal likelihood of heathrow data
    Y2[t] ~ dnorm(Y2[t-1], tau10)                  # normal prediction model
    rmse.Y[t] <- sqrt(sum((mean.Y - Y2[t])^2 / N2))   # root mean square prediction error
  }
  # priors on measurement error and white noise variances
  tau9 ~ dgamma(0.001, 0.001)                  # gamma measurement error precision model
  sigma2_9 <- 1/tau9                           # measurement error variance
  tau10 ~ dgamma(0.001, 0.001)                 # gamma estimate error precision model
  sigma2_10 <- 1/tau10                         # estimate error variance
}

# Specify initial values
inits.heathrow9 <- list("Y2" = c(rep.int(NA,1461), rep.int(mean.heathrow2,7)), "is.na(Heathrow)" = mean.heathrow2)
inits.heathrow10 <- list("Y2" = c(rep.int(NA,1461), rep.int(21,7)), "is.na(Heathrow)" = mean.heathrow2)
jags.inits.heathrow5 <- list(inits.heathrow9, inits.heathrow10)

# Monitor the parameters to be used for the prediction
jags.param.heathrow5 <- c("sigma2_9","sigma2_10","Y2", "rmse.Y")

# Fitting the new model
jags.mod.fit.heathrow5 <- jags(data = jags.data.heathrow5, inits = jags.inits.heathrow5,
                      parameters.to.save = jags.param.heathrow5, n.chains = 2,
                      n.iter = 10000, n.burnin = 5000, n.thin = 1,
                      model.file = jags.mod.heathrow5)

# Get point and interval estimates
print(jags.mod.fit.heathrow5)
# Inference for Bugs model at
"/var/folders/pc/hksslngn6_56y6dc8464zthm0000gn/T//Rtmp2LnJkG/model2b97f59f405.txt", fit using jags,
# 2 chains, each with 10000 iterations (first 5000 discarded)
# n.sims = 10000 iterations saved
#            mu.vect sd.vect     2.5%     25%      50%      75%    97.5%  Rhat  n.eff
# Y2[1]       10.790   6.303   -1.534    6.539   10.818   15.083   23.244 1.001 10000
# Y2[2]       10.975   3.993    3.218    8.242   10.947   13.658   18.906 1.001 10000
# Y2[3]       10.873   3.553    4.004    8.462   10.851   13.272   17.899 1.001  3400
# Y2[4]       13.037   3.427    6.420   10.720   13.035   15.326   19.835 1.001 10000
# Y2[5]       17.001   3.366   10.411   14.739   17.026   19.241   23.584 1.001  4900
# Y2[6]       18.592   3.359   11.825   16.375   18.608   20.874   25.124 1.001 10000
# Y2[7]       20.215   3.394   13.554   17.915   20.218   22.530   26.790 1.001  3600
# Y2[8]       21.225   3.470   14.422   18.920   21.211   23.622   27.871 1.003   750
# Y2[9]       26.892   3.626   19.741   24.580   26.972   29.321   33.790 1.001 10000
# Y2[10]      29.851   3.957   21.883   27.328   29.949   32.486   37.174 1.001  6200
# For each parameter, n.eff is a crude measure of effective sample size,
# and Rhat is the potential scale reduction factor (at convergence, Rhat=1).

# DIC info (using the rule, pD = var(deviance)/2)
# pD = 17765.9 and DIC = 24324.0
# DIC is an estimate of expected predictive error (lower deviance is better).


# plot density manually
sim.values.heathrow5 <- jags.mod.fit.heathrow5$BUGSoutput$sims.list
```

```
df.heathrow9 <- data.frame(sigma2_9 = sim.values.heathrow5$sigma2_9,
                           sigma2_10 = sim.values.heathrow5$sigma2_10)

heathrow_sigma2_9 <- ggplot(data = df.heathrow9, aes(x = sigma2_9)) +
  geom_density(colour = '#0093af', fill = '#0093af', alpha = 0.3) +
  theme(axis.title = element_blank(),
        axis.text = element_text(size = 12),
        plot.title = element_text(size = 14)) +
  geom_vline(xintercept = 0, linetype = "dashed",
             colour = "#aa0078", size = 1) +
  ggtitle('Posterior density of sigma2_9 - measurement variance')
heathrow_sigma2_9

heathrow_sigma2_10 <- ggplot(data = df.heathrow9, aes(x = sigma2_10)) +
  geom_density(colour = '#0093af', fill = '#0093af', alpha = 0.3) +
  theme(axis.title = element_blank(),
        axis.text = element_text(size = 12),
        plot.title = element_text(size = 14)) +
  geom_vline(xintercept = 0, linetype = "dashed",
             colour = "#aa0078", size = 1) +
  ggtitle('Posterior density of sigma2_10 - estimate variance')
heathrow_sigma2_10


# Create an MCMC object from the output of the heathrow model
jags.mcmc.heathrow5 <- as.mcmc(jags.mod.fit.heathrow5)
# Graphical parameters
par(mar = c(2,4,4,2), cex = 1.0)
layout(matrix(c(1,1,2,2),2,2,byrow=TRUE))
# for a traceplot, producing for groups of traceplots
traceplot(jags.mcmc.heathrow5[,101:150], params = c("sigma2_9", "sigma2_10", "Y2"))
traceplot(jags.mcmc.heathrow5[,1:10], params = c("sigma2_9", "sigma2_10", "Y2"))
traceplot(jags.mcmc.heathrow5[,301:350], params = c("sigma2_9", "sigma2_10", "Y2"))
traceplot(jags.mcmc.heathrow5[,1551:1600], params = c("sigma2_9", "sigma2_10", "Y2"))
traceplot(jags.mcmc.heathrow5[,1466:1475], params = c("sigma2_9", "sigma2_10", "Y2"))
traceplot(jags.mcmc.heathrow5[,1751:1800], params = c("sigma2_9", "sigma2_10", "Y2"))

# for summary statistics
summary(jags.mcmc.heathrow5)
# Iterations = 5001:10000
# Thinning interval = 1
# Number of chains = 2
# Sample size per chain = 5000

# 1. Empirical mean and standard deviation for each variable, plus standard error of the mean:
#                Mean       SD   Naive SE  Time-series SE
# deviance     6558.02873 189.94827 1.8994827      1.523e+01
# Parameters with associated measurements
# rmse.Y[1090]    0.10362   0.07138 0.0007138     9.640e-04
# rmse.Y[1091]    0.12347   0.07702 0.0007702     1.086e-03
# rmse.Y[1092]    0.09997   0.06913 0.0006913     9.502e-04
# rmse.Y[1093]    0.13923   0.08119 0.0008119     1.263e-03
# rmse.Y[1094]    0.15820   0.08445 0.0008445     1.505e-03
# rmse.Y[1095]    0.10579   0.07265 0.0007265     1.244e-03
# rmse.Y[1096]    0.09203   0.06663 0.0006663     9.706e-04
# rmse.Y[1097]    0.36323   0.10049 0.0010049     4.167e-03
```

```
# rmse.Y[1098]    0.33372   0.09824 0.0009824     3.327e-03
# rmse.Y[1099]    0.10864   0.08005 0.0008005     2.287e-03
# Parameters with missing data
# rmse.Y[1100]    0.14270   0.10926 0.0010926     3.527e-03
# rmse.Y[1101]    0.16902   0.13029 0.0013029     4.953e-03
# rmse.Y[1102]    0.19196   0.14798 0.0014798     6.667e-03
# rmse.Y[1103]    0.20942   0.16159 0.0016159     8.363e-03
# rmse.Y[1104]    0.22420   0.17821 0.0017821     1.013e-02
# rmse.Y[1105]    0.24081   0.19248 0.0019248     1.148e-02
# rmse.Y[1106]    0.25910   0.20773 0.0020773     1.364e-02
# rmse.Y[1107]    0.27752   0.22109 0.0022109     1.596e-02
# rmse.Y[1108]    0.29614   0.23462 0.0023462     1.936e-02
# rmse.Y[1109]    0.31467   0.24772 0.0024772     2.118e-02
# rmse.Y[1110]    0.32937   0.25785 0.0025785     2.274e-02

# 2. Quantiles for each variable:
#               2.5%       25%       50%       75%    97.5%
# deviance     6.250e+03 6455.18939  6.551e+03 6644.8565 6870.3844
# Parameters with associated measurements
# rmse.Y[1090]  4.413e-03   0.04494  9.264e-02   0.1503   0.2637
# rmse.Y[1091]  6.400e-03   0.06243  1.155e-01   0.1747   0.2898
# rmse.Y[1092]  4.584e-03   0.04412  8.912e-02   0.1445   0.2563
# rmse.Y[1093]  8.722e-03   0.07671  1.335e-01   0.1934   0.3104
# rmse.Y[1094]  1.238e-02   0.09519  1.548e-01   0.2165   0.3313
# rmse.Y[1095]  4.686e-03   0.04623  9.545e-02   0.1527   0.2686
# rmse.Y[1096]  3.489e-03   0.03822  7.975e-02   0.1338   0.2457
# rmse.Y[1097]  1.596e-01   0.29639  3.649e-01   0.4317   0.5543
# rmse.Y[1098]  1.398e-01   0.26942  3.344e-01   0.3996   0.5236
# rmse.Y[1099]  4.403e-03   0.04498  9.295e-02   0.1567   0.2999
# Parameters with missing data
# rmse.Y[1100]  5.614e-03   0.05793  1.187e-01   0.2044   0.4110
# rmse.Y[1101]  7.462e-03   0.06607  1.409e-01   0.2419   0.4861
# rmse.Y[1102]  7.636e-03   0.07610  1.602e-01   0.2757   0.5492
# rmse.Y[1103]  8.643e-03   0.08297  1.733e-01   0.3014   0.6012
# rmse.Y[1104]  8.708e-03   0.08603  1.853e-01   0.3231   0.6769
# rmse.Y[1105]  8.887e-03   0.09165  1.985e-01   0.3427   0.7387
# rmse.Y[1106]  9.708e-03   0.09843  2.115e-01   0.3708   0.7921
# rmse.Y[1107]  1.128e-02   0.10832  2.261e-01   0.3947   0.8313
# rmse.Y[1108]  1.177e-02   0.11538  2.438e-01   0.4172   0.8830
# rmse.Y[1109]  1.165e-02   0.12316  2.621e-01   0.4447   0.9321
# rmse.Y[1110]  1.343e-02   0.12822  2.759e-01   0.4681   0.9662


# Produce an MCMC trace of sigma2_9 output
MCMCtrace(jags.mcmc.heathrow5,
        params = c('sigma2_9'),
        type = 'density',
        ind = TRUE,
        ISB = FALSE,
        pdf = FALSE,
        col_den = c("#aa0078", "#0093af"))

# Produce an MCMC trace of sigma2_10 output
MCMCtrace(jags.mcmc.heathrow5,
        params = c('sigma2_10'),
        type = 'density',
        ind = TRUE,
```

```
        ISB = FALSE,
        pdf = FALSE,
        col_den = c("#aa0078", "#0093af"))

# Produce an MCMC trace of rmse.Y output
MCMCtrace(jags.mcmc.heathrow5,
        params = c('rmse.Y'),
        type = 'density',
        ind = TRUE,
        ISB = FALSE,
        pdf = FALSE,
        col_den = c("#aa0078", "#0093af"))

gelman.diag(jags.mcmc.heathrow5)
# Potential scale reduction factors:
#          Point est. Upper C.I.
# alpha6       1.55       4.48
# beta6        1.55       4.47

# Multivariate psrf
# 1.29

kbl(gelman.diag(jags.mcmc.heathrow5)) %>% kable_styling()
gelman.plot(jags.mcmc.heathrow5)


# heathrow data for 2000-2003 and first week of 2004
heathrow4b <- heathrow4[2:1468,]
tail(heathrow4b)

# Plotting the measurements, estimates and rmse for 2000-2003 and first week of 2004 for RW model 1 with rmse.Y
mu.rmse.heathrow6 <- jags.mod.fit.heathrow5$BUGSoutput$mean$rmse.Y
sd.rmse.heathrow6 <- jags.mod.fit.heathrow5$BUGSoutput$sd$rmse.Y

# ```{r}
df.heathrow10 <- data.frame(x6 = heathrow4b$Date, y11 = mu.rmse.heathrow6,
                            lower.rmse.heathrow6 = mu.rmse.heathrow6 - 1.96*sd.rmse.heathrow6,
                            upper.rmse.heathrow6 = mu.rmse.heathrow6 + 1.96*sd.rmse.heathrow6)

ggplot(data = df.heathrow10) +
  geom_point(aes(x = x6, y = y11), colour = 'grey31', size = 2) +
  geom_line(aes(x = x6, y = y11), colour = '#0093af', size = 1) +
  geom_line(aes(x = x6, y = lower.rmse.heathrow6), linetype = "dashed", colour = "#aa0078", size = 1) +
  geom_line(aes(x = x6, y = upper.rmse.heathrow6), linetype = "dashed", colour = "#aa0078", size = 1) +
  xlab('Date') + ylab('Root mean square prediction values of PM10 at Heathrow') +
  ggtitle('PM10 rmse at Heathrow for 2000-2003 and first week of 2004 with RW model 1') +
  theme(axis.title = element_text(size = 14),
        axis.text = element_text(size = 12),
        plot.title = element_text(size = 14))




# RW2 model with root mean square

# List the data to be used
jags.data.heathrow6 <- list("Heathrow2", "N2")
```

```
# Model 2 with rmse
jags.mod.heathrow6 <- function(){
  Y2[1] ~ dnorm(0, 0.001)
  Y2[2] ~ dnorm(0, 0.001)
  mean.Y <- 19.58195
  for(t in 3:N2){
    Heathrow2[t] ~ dnorm(Y2[t], tau11)             # normal likelihood of heathrow data
    Y2[t] ~ dnorm((2*Y2[t-1] - Y2[t-2]), tau12)      # normal prediction model
    rmse.Y[t] <- sqrt(sum((mean.Y - Y2[t])^2 / N2))   # root mean square prediction error
  }
  # priors on measurement error and white noise variances
  tau11 ~ dgamma(0.001, 0.001)                 # gamma measurement error precision model
  sigma2_11 <- 1/tau11                       # measurement error variance
  tau12 ~ dgamma(0.001, 0.001)                 # gamma estimate error precision model
  sigma2_12 <- 1/tau12                       # estimate error variance
}

# Specify initial values
inits.heathrow11 <- list("Y2" = c(rep.int(NA,1461), rep.int(mean.heathrow2,7)), "is.na(Heathrow)" =
mean.heathrow2)
inits.heathrow12 <- list("Y2" = c(rep.int(NA,1461), rep.int(21,7)), "is.na(Heathrow)" = mean.heathrow2)
jags.inits.heathrow6 <- list(inits.heathrow11, inits.heathrow12)

# Monitor the parameters to be used for the prediction
jags.param.heathrow6 <- c("sigma2_11","sigma2_12","Y2", "rmse.Y")

# Fitting the new model
jags.mod.fit.heathrow6 <- jags(data = jags.data.heathrow6, inits = jags.inits.heathrow6,
                      parameters.to.save = jags.param.heathrow6, n.chains = 2,
                      n.iter = 10000, n.burnin = 5000, n.thin = 1,
                      model.file = jags.mod.heathrow6)

# Get point and interval estimates
print(jags.mod.fit.heathrow6)
# Inference for Bugs model at
"/var/folders/pc/hkssIngn6_56y6dc8464zthm0000gn/T//Rtmp2LnJkG/model2b94157694a.txt", fit using jags,
# 2 chains, each with 10000 iterations (first 5000 discarded)
# n.sims = 10000 iterations saved
# mu.vect sd.vect    2.5%     25%     50%     75%   97.5%   Rhat n.eff
# Y2[1]         5.879   2.838   0.014   4.205   6.458   7.879   10.309  1.764    5
# Y2[2]         6.113   2.481   1.120   4.678   6.697   7.850   10.012  1.978    4
# Y2[3]         6.346   2.196   1.860   5.017   6.933   7.880    9.900  2.223    3
# Y2[4]         6.579   1.982   2.297   5.326   7.168   7.976    9.763  2.473    3
# Y2[5]         6.806   1.837   2.211   5.637   7.320   8.141    9.666  2.663    3
# Y2[6]         7.028   1.749   2.134   5.997   7.289   8.387    9.691  2.743    3
# Y2[7]         7.250   1.694   2.341   6.263   7.191   8.623    9.755  2.731    3
# Y2[8]         7.476   1.658   2.891   6.472   7.276   8.830    9.950  2.001    4
# Y2[9]         7.708   1.626   3.698   6.668   7.474   9.067   10.261  2.200    4
# Y2[10]        7.946   1.599   4.675   6.853   7.691   9.297   10.627  2.267    3
# For each parameter, n.eff is a crude measure of effective sample size,
# and Rhat is the potential scale reduction factor (at convergence, Rhat=1).

# DIC info (using the rule, pD = var(deviance)/2)
# pD = 2368.3 and DIC = 11409.2
# DIC is an estimate of expected predictive error (lower deviance is better).
```

```
# plot density manually
sim.values.heathrow6 <- jags.mod.fit.heathrow6$BUGSoutput$sims.list

df.heathrow11 <- data.frame(sigma2_11 = sim.values.heathrow6$sigma2_11,
                            sigma2_12 = sim.values.heathrow6$sigma2_12)

heathrow_sigma2_11 <- ggplot(data = df.heathrow11, aes(x = sigma2_11)) +
  geom_density(colour = '#0093af', fill = '#0093af', alpha = 0.3) +
  theme(axis.title = element_blank(),
        axis.text = element_text(size = 12),
        plot.title = element_text(size = 14)) +
  geom_vline(xintercept = 0, linetype = "dashed",
             colour = "#aa0078", size = 1) +
  ggtitle('Posterior density of sigma2_11 - measurement variance')
heathrow_sigma2_11

heathrow_sigma2_12 <- ggplot(data = df.heathrow11, aes(x = sigma2_12)) +
  geom_density(colour = '#0093af', fill = '#0093af', alpha = 0.3) +
  theme(axis.title = element_blank(),
        axis.text = element_text(size = 12),
        plot.title = element_text(size = 14)) +
  geom_vline(xintercept = 0, linetype = "dashed",
             colour = "#aa0078", size = 1) +
  ggtitle('Posterior density of sigma2_12 - estimate variance')
heathrow_sigma2_12


# Create an MCMC object from the output of the heathrow model
jags.mcmc.heathrow6 <- as.mcmc(jags.mod.fit.heathrow6)
# Graphical parameters
par(mar = c(2,4,4,2), cex = 1.0)
layout(matrix(c(1,1,2,2),2,2,byrow=TRUE))
# for a traceplot, producing for groups of traceplots
traceplot(jags.mcmc.heathrow6[,101:150], params = c("sigma2_11", "sigma2_12", "Y2", "rmse.Y"))
traceplot(jags.mcmc.heathrow6[,1:10], params = c("sigma2_11", "sigma2_12", "Y2", "rmse.Y"))
traceplot(jags.mcmc.heathrow6[,251:300], params = c("sigma2_11", "sigma2_12", "Y2", "rmse.Y"))
traceplot(jags.mcmc.heathrow6[,1551:1600], params = c("sigma2_11", "sigma2_12", "Y2", "rmse.Y"))
traceplot(jags.mcmc.heathrow6[,1466:1475], params = c("sigma2_11", "sigma2_12", "Y2", "rmse.Y"))
traceplot(jags.mcmc.heathrow6[,1731:1780], params = c("sigma2_11", "sigma2_12", "Y2", "rmse.Y"))

# for summary statistics
summary(jags.mcmc.heathrow6)
# Iterations = 5001:10000
# Thinning interval = 1
# Number of chains = 2
# Sample size per chain = 5000

# 1. Empirical mean and standard deviation for each variable, plus standard error of the mean:
#                Mean       SD   Naive SE  Time-series SE
# deviance     9.041e+03 74.30139 0.7430139      39.592119
# Parameters with associated measurements
# rmse.Y[1090]  3.058e-01  0.03109 0.0003109      0.006577
# rmse.Y[1091]  3.124e-01  0.02982 0.0002982      0.005105
# rmse.Y[1092]  3.194e-01  0.03068 0.0003068      0.004029
# rmse.Y[1093]  3.270e-01  0.03401 0.0003401      0.003728
# rmse.Y[1094]  3.354e-01  0.03921 0.0003921      0.004037
```

```
# rmse.Y[1095]  3.446e-01  0.04535 0.0004535      0.005070
# rmse.Y[1096]  3.548e-01  0.05176 0.0005176      0.006615
# rmse.Y[1097]  3.661e-01  0.05789 0.0005789      0.008731
# rmse.Y[1098]  3.785e-01  0.06319 0.0006319      0.010749
# rmse.Y[1099]  3.919e-01  0.06758 0.0006758      0.011725
# Parameters with missing data
# rmse.Y[1100]  4.058e-01  0.07093 0.0007093      0.013913
# rmse.Y[1101]  4.199e-01  0.07319 0.0007319      0.014759
# rmse.Y[1102]  4.340e-01  0.07453 0.0007453      0.015326
# rmse.Y[1103]  4.477e-01  0.07484 0.0007484      0.014600
# rmse.Y[1104]  4.608e-01  0.07391 0.0007391      0.014008
# rmse.Y[1105]  4.730e-01  0.07165 0.0007165      0.012066
# rmse.Y[1106]  4.843e-01  0.06822 0.0006822      0.012753
# rmse.Y[1107]  4.949e-01  0.06390 0.0006390      0.011776
# rmse.Y[1108]  5.046e-01  0.05900 0.0005900      0.012051
# rmse.Y[1109]  5.133e-01  0.05384 0.0005384      0.013944

# 2. Quantiles for each variable:
#               2.5%       25%       50%       75%     97.5%
# deviance     8.952e+03  8.983e+03  9.014e+03 9086.97755 9208.87575
# Parameters with associated measurements
# rmse.Y[1090]  2.689e-01  2.856e-01  2.989e-01   0.31719   0.39747
# rmse.Y[1091]  2.731e-01  2.909e-01  3.082e-01   0.32674   0.39104
# rmse.Y[1092]  2.764e-01  2.929e-01  3.158e-01   0.34046   0.38653
# rmse.Y[1093]  2.783e-01  2.949e-01  3.229e-01   0.35604   0.39122
# rmse.Y[1094]  2.789e-01  2.988e-01  3.308e-01   0.36962   0.40505
# rmse.Y[1095]  2.798e-01  3.035e-01  3.366e-01   0.38625   0.42360
# rmse.Y[1096]  2.812e-01  3.087e-01  3.426e-01   0.40456   0.44346
# rmse.Y[1097]  2.840e-01  3.148e-01  3.523e-01   0.42388   0.46416
# rmse.Y[1098]  2.893e-01  3.225e-01  3.636e-01   0.44210   0.48571
# rmse.Y[1099]  2.976e-01  3.309e-01  3.769e-01   0.45957   0.50798
# Parameters with missing data
# rmse.Y[1100]  3.069e-01  3.395e-01  3.911e-01   0.47475   0.53001
# rmse.Y[1101]  3.183e-01  3.498e-01  4.065e-01   0.48882   0.54997
# rmse.Y[1102]  3.286e-01  3.604e-01  4.224e-01   0.50293   0.56721
# rmse.Y[1103]  3.376e-01  3.733e-01  4.424e-01   0.51563   0.57886
# rmse.Y[1104]  3.490e-01  3.873e-01  4.618e-01   0.52797   0.58462
# rmse.Y[1105]  3.589e-01  4.024e-01  4.814e-01   0.53693   0.58665
# rmse.Y[1106]  3.698e-01  4.167e-01  4.985e-01   0.54409   0.58703
# rmse.Y[1107]  3.827e-01  4.309e-01  5.133e-01   0.54820   0.58886
# rmse.Y[1108]  3.981e-01  4.462e-01  5.255e-01   0.54940   0.59227
# rmse.Y[1109]  4.145e-01  4.607e-01  5.325e-01   0.55257   0.59661



# Produce an MCMC trace of sigma2_11 output
MCMCtrace(jags.mcmc.heathrow6,
        params = c('sigma2_11'),
        type = 'density',
        ind = TRUE,
        ISB = FALSE,
        pdf = FALSE,
        col_den = c("#aa0078", "#0093af"))

# Produce an MCMC trace of sigma2_12 output
MCMCtrace(jags.mcmc.heathrow6,
        params = c('sigma2_12'),
        type = 'density',
```

```
        ind = TRUE,
        ISB = FALSE,
        pdf = FALSE,
        col_den = c("#aa0078", "#0093af"))

# Produce an MCMC trace of rmse.Y output
MCMCtrace(jags.mcmc.heathrow6,
        params = c('rmse.Y'),
        type = 'density',
        ind = TRUE,
        ISB = FALSE,
        pdf = FALSE,
        col_den = c("#aa0078", "#0093af"))

gelman.diag(jags.mcmc.heathrow6)
# Potential scale reduction factors:
#          Point est. Upper C.I.
# alpha6      1.55      4.48
# beta6       1.55      4.47

# Multivariate psrf
# 1.29

kbl(gelman.diag(jags.mcmc.heathrow6)) %>% kable_styling()
gelman.plot(jags.mcmc.heathrow6)


# heathrow data for 2000-2003 and first week of 2004
heathrow4c <- heathrow4[3:1468,]
tail(heathrow4c)

# Plotting the measurements, estimates and rmse for 2000-2003 and first week of 2004 for RW model 1 with rmse.Y
mu.rmse.heathrow7 <- jags.mod.fit.heathrow6$BUGSoutput$mean$rmse.Y
sd.rmse.heathrow7 <- jags.mod.fit.heathrow6$BUGSoutput$sd$rmse.Y

# ```{r}
df.heathrow12 <- data.frame(x7 = heathrow4c$Date, y12 = mu.rmse.heathrow7,
                    lower.rmse.heathrow7 = mu.rmse.heathrow7 - 1.96*sd.rmse.heathrow7,
                    upper.rmse.heathrow7 = mu.rmse.heathrow7 + 1.96*sd.rmse.heathrow7)

ggplot(data = df.heathrow12) +
  geom_point(aes(x = x7, y = y12), colour = 'grey31', size = 2) +
  geom_line(aes(x = x7, y = y12), colour = '#0093af', size = 1) +
  geom_line(aes(x = x7, y = lower.rmse.heathrow7), linetype = "dashed", colour = "#aa0078", size = 1) +
  geom_line(aes(x = x7, y = upper.rmse.heathrow7), linetype = "dashed", colour = "#aa0078", size = 1) +
  xlab('Date') + ylab('Root mean square prediction values of PM10 at Heathrow') +
  ggtitle('PM10 rmse at Heathrow for 2000-2003 and first week of 2004 with RW model 2') +
  theme(axis.title = element_text(size = 14),
        axis.text = element_text(size = 12),
        plot.title = element_text(size = 14))



# Y vs Heathrow

# heathrow data for just 2000-2003 and first week in 2004 with root mean square function
heathrow4 <- heathrow2 %>% slice(1:1468)
```

```
tail(heathrow4)
heathrow4[1001:1100,]
heathrow4[1101:1200,]
heathrow4[1201:1300,]
heathrow4[1450:1470,]
# Set seed for reproducibility
set.seed(234)
# Set N to be the length of the data
N2 <- 1468
# Mean of Heathrow data
mean.heathrow2 <- mean(heathrow4$Heathrow, na.rm = TRUE)
mean.heathrow2
# Extract Heathrow column data
Heathrow2 <- heathrow4$Heathrow
tail(Heathrow2)

# RW1 model with root mean square

# List the data to be used
jags.data.heathrow7 <- list("Heathrow2", "N2")

# Model
jags.mod.heathrow7 <- function(){
  Y2[1] ~ dnorm(0, 0.001)
  for(t in 2:N2){
    Heathrow2[t] ~ dnorm(Y2[t], tau21)                  # normal likelihood of heathrow data
    Y2[t] ~ dnorm(Y2[t-1], tau22)                       # normal prediction model
    rmse.Y[t] <- sqrt(sum((Heathrow2[t] - Y2[t])^2 / N2))   # root mean square prediction error
  }
  # priors on measurement error and white noise variances
  tau21 ~ dgamma(0.001, 0.001)                  # gamma measurement error precision model
  sigma2_21 <- 1/tau21                          # measurement error variance
  tau22 ~ dgamma(0.001, 0.001)                  # gamma estimate error precision model
  sigma2_22 <- 1/tau22                          # estimate error variance
}

# Specify initial values
inits.heathrow13 <- list("Y2" = c(rep.int(NA,1461), rep.int(mean.heathrow2,7)), "is.na(Heathrow)" =
mean.heathrow2)
inits.heathrow14 <- list("Y2" = c(rep.int(NA,1461), rep.int(21,7)), "is.na(Heathrow)" = mean.heathrow2)
jags.inits.heathrow7 <- list(inits.heathrow13, inits.heathrow14)

# Monitor the parameters to be used for the prediction
jags.param.heathrow7 <- c("sigma2_21","sigma2_22","Y2", "rmse.Y")

# Fitting the new model
jags.mod.fit.heathrow7 <- jags(data = jags.data.heathrow7, inits = jags.inits.heathrow7,
                      parameters.to.save = jags.param.heathrow7, n.chains = 2,
                      n.iter = 10000, n.burnin = 5000, n.thin = 1,
                      model.file = jags.mod.heathrow7)

# Get point and interval estimates
print(jags.mod.fit.heathrow7)
# Inference for Bugs model at
"/var/folders/pc/hkssIngn6_56y6dc8464zthm0000gn/T//RtmpSTiK0N/model24a3c284f87.txt", fit using jags,
# 2 chains, each with 10000 iterations (first 5000 discarded)
# n.sims = 10000 iterations saved
```

```
#            mu.vect sd.vect    2.5%     25%     50%     75%   97.5%  Rhat n.eff
# Y2[1]       10.746   6.195  -1.571   6.555  10.706  14.995  22.660 1.001 10000
# Y2[2]       10.977   3.975   3.231   8.315  10.946  13.643  18.885 1.001 10000
# Y2[3]       10.887   3.533   3.919   8.521  10.873  13.279  17.786 1.001  6500
# Y2[4]       13.118   3.454   6.286  10.868  13.126  15.448  19.780 1.001 10000
# Y2[5]       17.084   3.396  10.304  14.846  17.114  19.351  23.599 1.001 10000
# Y2[6]       18.640   3.436  11.805  16.329  18.738  20.949  25.274 1.001 10000
# Y2[7]       20.207   3.445  13.437  17.863  20.239  22.551  26.889 1.001  7100
# Y2[8]       21.266   3.411  14.483  19.019  21.288  23.567  27.902 1.006   380
# Y2[9]       26.883   3.607  19.725  24.630  26.947  29.273  33.650 1.005  1600
# Y2[10]      29.841   3.956  22.158  27.321  29.916  32.460  37.313 1.008  1500
# For each parameter, n.eff is a crude measure of effective sample size,
# and Rhat is the potential scale reduction factor (at convergence, Rhat=1).

# DIC info (using the rule, pD = var(deviance)/2)
# pD = 19661.9 and DIC = 26211.7
# DIC is an estimate of expected predictive error (lower deviance is better).



# plot density manually
sim.values.heathrow7 <- jags.mod.fit.heathrow7$BUGSoutput$sims.list

df.heathrow13 <- data.frame(sigma2_21 = sim.values.heathrow7$sigma2_21,
                            sigma2_22 = sim.values.heathrow7$sigma2_22)

heathrow_sigma2_21 <- ggplot(data = df.heathrow13, aes(x = sigma2_21)) +
  geom_density(colour = '#0093af', fill = '#0093af', alpha = 0.3) +
  theme(axis.title = element_blank(),
        axis.text = element_text(size = 12),
        plot.title = element_text(size = 14)) +
  geom_vline(xintercept = 0, linetype = "dashed",
             colour = "#aa0078", size = 1) +
  ggtitle('Posterior density of sigma2_21 - measurement variance')
heathrow_sigma2_21

heathrow_sigma2_22 <- ggplot(data = df.heathrow13, aes(x = sigma2_22)) +
  geom_density(colour = '#0093af', fill = '#0093af', alpha = 0.3) +
  theme(axis.title = element_blank(),
        axis.text = element_text(size = 12),
        plot.title = element_text(size = 14)) +
  geom_vline(xintercept = 0, linetype = "dashed",
             colour = "#aa0078", size = 1) +
  ggtitle('Posterior density of sigma2_22 - estimate variance')
heathrow_sigma2_22



# Create an MCMC object from the output of the heathrow model
jags.mcmc.heathrow7 <- as.mcmc(jags.mod.fit.heathrow7)
# Graphical parameters
par(mar = c(2,4,4,2), cex = 1.0)
layout(matrix(c(1,1,2,2),2,2,byrow=TRUE))
# for a traceplot, producing for groups of traceplots
traceplot(jags.mcmc.heathrow7[,101:150], params = c("sigma2_21", "sigma2_22", "Y2"))
traceplot(jags.mcmc.heathrow7[,1:10], params = c("sigma2_21", "sigma2_22", "Y2"))
traceplot(jags.mcmc.heathrow7[,251:300], params = c("sigma2_21", "sigma2_22", "Y2"))
traceplot(jags.mcmc.heathrow7[,1551:1600], params = c("sigma2_21", "sigma2_22", "Y2"))
traceplot(jags.mcmc.heathrow7[,1466:1475], params = c("sigma2_21", "sigma2_22", "Y2"))
```

```
traceplot(jags.mcmc.heathrow7[,1751:1800], params = c("sigma2_21", "sigma2_22", "Y2"))

# for summary statistics
summary(jags.mcmc.heathrow7)
# Iterations = 5001:10000
# Thinning interval = 1
# Number of chains = 2
# Sample size per chain = 5000

# 1. Empirical mean and standard deviation for each variable, plus standard error of the mean:
#                Mean       SD   Naive SE  Time-series SE
# deviance     6.550e+03 199.48946 1.9948946     1.706e+01
# Parameters with associated measurements
# rmse.Y[1090]  7.164e-02   0.05467 0.0005467     6.151e-04
# rmse.Y[1091]  8.912e-02   0.06517 0.0006517     8.597e-04
# rmse.Y[1092]  1.039e-01   0.07082 0.0007082     9.790e-04
# rmse.Y[1093]  7.447e-02   0.05675 0.0005675     7.456e-04
# rmse.Y[1094]  1.116e-01   0.07424 0.0007424     1.370e-03
# rmse.Y[1095]  1.154e-01   0.07654 0.0007654     1.295e-03
# rmse.Y[1096]  1.497e-01   0.08279 0.0008279     1.329e-03
# rmse.Y[1097]  3.637e-01   0.10917 0.0010917     5.642e-03
# rmse.Y[1098]  2.528e-01   0.10715 0.0010715     5.495e-03
# rmse.Y[1099]  2.571e-01   0.10779 0.0010779     4.377e-03
# Parameters with missing data
# rmse.Y[1100]  1.097e-01   0.08332 0.0008332     9.141e-04
# rmse.Y[1101]  1.086e-01   0.08311 0.0008311     8.517e-04
# rmse.Y[1102]  1.098e-01   0.08372 0.0008372     1.171e-03
# rmse.Y[1103]  1.090e-01   0.08337 0.0008337     9.384e-04
# rmse.Y[1104]  1.086e-01   0.08553 0.0008553     9.035e-04
# rmse.Y[1105]  1.090e-01   0.08429 0.0008429     1.065e-03
# rmse.Y[1106]  1.099e-01   0.08437 0.0008437     1.174e-03
# rmse.Y[1107]  1.098e-01   0.08346 0.0008346     9.201e-04
# rmse.Y[1108]  1.080e-01   0.08311 0.0008311     1.024e-03
# rmse.Y[1109]  1.080e-01   0.08315 0.0008315     9.179e-04

# 2. Quantiles for each variable:
#                2.5%       25%       50%       75%     97.5%
# deviance     6.220e+03  6.446e+03 6545.74100 6640.1405 6836.3138
# Parameters with associated measurements
# rmse.Y[1090]  3.089e-03  2.888e-02   0.06054   0.1023   0.2032
# rmse.Y[1091]  3.759e-03  3.696e-02   0.07669   0.1277   0.2444
# rmse.Y[1092]  4.578e-03  4.681e-02   0.09298   0.1501   0.2625
# rmse.Y[1093]  2.647e-03  2.956e-02   0.06264   0.1072   0.2123
# rmse.Y[1094]  5.791e-03  5.200e-02   0.10094   0.1607   0.2762
# rmse.Y[1095]  5.325e-03  5.339e-02   0.10643   0.1658   0.2829
# rmse.Y[1096]  9.491e-03  8.704e-02   0.14550   0.2065   0.3222
# rmse.Y[1097]  1.683e-01  2.911e-01   0.36050   0.4281   0.5860
# rmse.Y[1098]  5.737e-02  1.823e-01   0.24896   0.3144   0.4744
# rmse.Y[1099]  4.505e-02  1.830e-01   0.25664   0.3314   0.4715
# Parameters with missing data
# rmse.Y[1100]  4.462e-03  4.408e-02   0.09349   0.1581   0.3120
# rmse.Y[1101]  4.539e-03  4.282e-02   0.09069   0.1574   0.3083
# rmse.Y[1102]  4.273e-03  4.425e-02   0.09256   0.1561   0.3103
# rmse.Y[1103]  4.258e-03  4.338e-02   0.09119   0.1569   0.3056
# rmse.Y[1104]  3.841e-03  4.089e-02   0.09023   0.1568   0.3195
# rmse.Y[1105]  4.180e-03  4.291e-02   0.09099   0.1576   0.3119
# rmse.Y[1106]  4.378e-03  4.324e-02   0.09153   0.1586   0.3116
```

```
# rmse.Y[1107]  4.139e-03  4.350e-02   0.09225   0.1585   0.3115
# rmse.Y[1108]  4.307e-03  4.290e-02   0.09003   0.1543   0.3101
# rmse.Y[1109]  4.266e-03  4.226e-02   0.09079   0.1550   0.3048


# Produce an MCMC trace of sigma2_21 output
MCMCtrace(jags.mcmc.heathrow7,
        params = c('sigma2_21'),
        type = 'density',
        ind = TRUE,
        ISB = FALSE,
        pdf = FALSE,
        col_den = c("#aa0078", "#0093af"))

# Produce an MCMC trace of sigma2_22 output
MCMCtrace(jags.mcmc.heathrow7,
        params = c('sigma2_22'),
        type = 'density',
        ind = TRUE,
        ISB = FALSE,
        pdf = FALSE,
        col_den = c("#aa0078", "#0093af"))

# Produce an MCMC trace of rmse.Y output
MCMCtrace(jags.mcmc.heathrow7,
        params = c('rmse.Y'),
        type = 'density',
        ind = TRUE,
        ISB = FALSE,
        pdf = FALSE,
        col_den = c("#aa0078", "#0093af"))


# Find the within-sequence variance (W) and see how close to 0 it is
gelman.heathrow7 <- gelman.diag(jags.mcmc.heathrow7, multivariate = FALSE)
W.heathrow7 <- gelman.heathrow7$W

# Investigate the eigenvalues for W
evals.W.heathrow7 <- eigen(W.heathrow7, only.values = TRUE)$values
min(evals.W.heathrow7)

# Indeed, increasing the tolerance shows that W is indeed positive definite.
matrixNormal::is.positive.definite(W.heathrow7, tol = 1e-18)

gelman.heathrow7
# Potential scale reduction factors:
#          Point est. Upper C.I.
# deviance       1.05       1.06
# sigma2_7       1.02       1.02
# sigma2_8       1.00       1.00
# Y2[1]          4.14       9.09
# Parameters with associated measurements
# Y2[1090]       1.29       2.14
# Y2[1091]       1.14       1.35
# Y2[1092]       1.11       1.16
# Y2[1093]       1.27       2.02
# Y2[1094]       1.86       3.69
```

```
# Y2[1095]      2.72      5.68
# Y2[1096]      3.65      7.73
# Y2[1097]      4.54      9.84
# Y2[1098]      5.39     12.12
# Y2[1099]      5.90     13.68
# Parameters with missing data
# Y2[1100]      5.93     14.22
# Y2[1101]      5.77     14.39
# Y2[1102]      5.58     14.51
# Y2[1103]      5.44     14.73
# Y2[1104]      5.34     14.87
# Y2[1105]      5.31     14.73
# Y2[1106]      5.29     14.11
# Y2[1107]      5.24     13.05
# Y2[1108]      5.16     11.90
# Y2[1109]      5.08     11.14

# Multivariate psrf
# 116

# Create a data.frame from the gelman.diag output to put into a kable table
rhat.heathrow7 <- as.data.frame((gelman.heathrow7)$psrf)
kbl(rhat.heathrow7) %>% kable_styling()

# Create a gelman plot of all the parameters and their densities
gelman.plot(jags.mcmc.heathrow7)


# heathrow data for 2000-2003 and first week of 2004
heathrow4b <- heathrow4[2:1468,]
tail(heathrow4b)

# Plotting the measurements, estimates and rmse for 2000-2003 and first week of 2004 for RW model 1 with rmse.Y
mu.rmse.heathrow8 <- jags.mod.fit.heathrow7$BUGSoutput$mean$rmse.Y
sd.rmse.heathrow8 <- jags.mod.fit.heathrow7$BUGSoutput$sd$rmse.Y

# ```{r}
df.heathrow14 <- data.frame(x8 = heathrow4b$Date, y13 = mu.rmse.heathrow8,
                            lower.rmse.heathrow8 = mu.rmse.heathrow8 - 1.96*sd.rmse.heathrow8,
                            upper.rmse.heathrow8 = mu.rmse.heathrow8 + 1.96*sd.rmse.heathrow8)

ggplot(data = df.heathrow14) +
  geom_point(aes(x = x8, y = y13), colour = 'grey31', size = 2) +
  geom_line(aes(x = x8, y = y13), colour = '#0093af', size = 1) +
  geom_line(aes(x = x8, y = lower.rmse.heathrow8), linetype = "dashed", colour = "#aa0078", size = 1) +
  geom_line(aes(x = x8, y = upper.rmse.heathrow8), linetype = "dashed", colour = "#aa0078", size = 1) +
  xlab('Date') + ylab('Root mean square prediction values of PM10 at Heathrow') +
  ggtitle('PM10 rmse at Heathrow for 2000-2003 and first week of 2004 with RW model 1') +
  theme(axis.title = element_text(size = 14),
        axis.text = element_text(size = 12),
        plot.title = element_text(size = 14))

# When comparing the Y estimates with the actual measurements (Heathrow) for RW1, where there are data the
rmse
# values have some variation but always remain above 0.02 and typically below 0.2. The 95% credible interval also
# has a generally negative lower interval boundary, which is not possible, and a upper interval boundary that
# typically does not get above 0.5. However, where there are no data, the rmse value remains relatively constant
```

```
# around a value of 0.09, with a constant credible interval (-0.05, 0.28). This means that when forcasting, the
# forecasted values will tend towards a constant value based on previous estimates and measurements and have an
# expanding credible interval in which the forecasted values would lie.



# RW2 model with root mean square

# List the data to be used
jags.data.heathrow8 <- list("Heathrow2", "N2")

# Model 2 with rmse
jags.mod.heathrow8 <- function(){
  Y2[1] ~ dnorm(0, 0.001)
  Y2[2] ~ dnorm(0, 0.001)
  for(t in 3:N2){
    Heathrow2[t] ~ dnorm(Y2[t], tau23)                  # normal likelihood of heathrow data
    Y2[t] ~ dnorm((2*Y2[t-1] - Y2[t-2]), tau24)          # normal prediction model
    rmse.Y[t] <- sqrt(sum((Heathrow2[t] - Y2[t])^2 / N2))   # root mean square prediction error
  }
  # priors on measurement error and white noise variances
  tau23 ~ dgamma(0.001, 0.001)                  # gamma measurement error precision model
  sigma2_23 <- 1/tau23                        # measurement error variance
  tau24 ~ dgamma(0.001, 0.001)                  # gamma estimate error precision model
  sigma2_24 <- 1/tau24                        # estimate error variance
}

# Specify initial values
inits.heathrow15 <- list("Y2" = c(rep.int(NA,1461), rep.int(mean.heathrow2,7)), "is.na(Heathrow)" =
mean.heathrow2)
inits.heathrow16 <- list("Y2" = c(rep.int(NA,1461), rep.int(21,7)), "is.na(Heathrow)" = mean.heathrow2)
jags.inits.heathrow8 <- list(inits.heathrow15, inits.heathrow16)

# Monitor the parameters to be used for the prediction
jags.param.heathrow8 <- c("sigma2_23","sigma2_24","Y2", "rmse.Y")

# Fitting the new model
jags.mod.fit.heathrow8 <- jags(data = jags.data.heathrow8, inits = jags.inits.heathrow8,
                      parameters.to.save = jags.param.heathrow8, n.chains = 2,
                      n.iter = 10000, n.burnin = 5000, n.thin = 1,
                      model.file = jags.mod.heathrow8)

# Get point and interval estimates
print(jags.mod.fit.heathrow8)
# Inference for Bugs model at
"/var/folders/pc/hksslngn6_56y6dc8464zthm0000gn/T//Rtmp5CJmE0/model20775f842ca0.txt", fit using jags,
# 2 chains, each with 10000 iterations (first 5000 discarded)
# n.sims = 10000 iterations saved
#           mu.vect sd.vect    2.5%     25%     50%     75%   97.5%  Rhat n.eff
# Y2[1]       8.310   4.322   0.659   4.525   8.988  11.359  15.852 2.399     3
# Y2[2]       8.256   3.531   1.821   5.148   8.864  10.645  14.489 2.290     3
# Y2[3]       8.201   2.792   2.899   5.780   8.708  10.078  13.226 2.084     4
# Y2[4]       8.147   2.133   4.054   6.430   8.404   9.557  12.271 1.717     5
# Y2[5]       8.100   1.608   4.980   6.983   8.042   9.292  11.420 1.223    12
# Y2[6]       8.063   1.319   5.677   7.161   7.905   9.083  10.625 1.009   190
# Y2[7]       8.034   1.352   5.268   7.065   8.106   9.010  10.265 1.651     5
# Y2[8]       8.010   1.619   4.689   6.837   8.301   9.390  10.327 2.630     3
```

```
# Y2[9]          7.998   1.972    4.154    6.382    8.568    9.735   10.697  3.370     3
# Y2[10]         7.991   2.325    3.717    5.917    8.387   10.115   11.179  3.845     2
# For each parameter, n.eff is a crude measure of effective sample size,
# and Rhat is the potential scale reduction factor (at convergence, Rhat=1).

# DIC info (using the rule, pD = var(deviance)/2)
# pD = 3180.9 and DIC = 12199.2
# DIC is an estimate of expected predictive error (lower deviance is better).


# plot density manually
sim.values.heathrow8 <- jags.mod.fit.heathrow8$BUGSoutput$sims.list

df.heathrow15 <- data.frame(sigma2_23 = sim.values.heathrow8$sigma2_23,
                            sigma2_24 = sim.values.heathrow8$sigma2_24)

heathrow_sigma2_23 <- ggplot(data = df.heathrow15, aes(x = sigma2_23)) +
  geom_density(colour = '#0093af', fill = '#0093af', alpha = 0.3) +
  theme(axis.title = element_blank(),
        axis.text = element_text(size = 12),
        plot.title = element_text(size = 14)) +
  geom_vline(xintercept = 0, linetype = "dashed",
             colour = "#aa0078", size = 1) +
  ggtitle('Posterior density of sigma2_23 - measurement variance')
heathrow_sigma2_23

heathrow_sigma2_24 <- ggplot(data = df.heathrow15, aes(x = sigma2_24)) +
  geom_density(colour = '#0093af', fill = '#0093af', alpha = 0.3) +
  theme(axis.title = element_blank(),
        axis.text = element_text(size = 12),
        plot.title = element_text(size = 14)) +
  geom_vline(xintercept = 0, linetype = "dashed",
             colour = "#aa0078", size = 1) +
  ggtitle('Posterior density of sigma2_24 - estimate variance')
heathrow_sigma2_24


# Create an MCMC object from the output of the heathrow model
jags.mcmc.heathrow8 <- as.mcmc(jags.mod.fit.heathrow8)
# Graphical parameters
par(mar = c(2,4,4,2), cex = 1.0)
layout(matrix(c(1,1,2,2),2,2,byrow=TRUE))
# for a traceplot, producing for groups of traceplots
traceplot(jags.mcmc.heathrow8[,101:150], params = c("sigma2_23", "sigma2_24", "Y2", "rmse.Y"))
traceplot(jags.mcmc.heathrow8[,1:10], params = c("sigma2_23", "sigma2_24", "Y2", "rmse.Y"))
traceplot(jags.mcmc.heathrow8[,251:300], params = c("sigma2_23", "sigma2_24", "Y2", "rmse.Y"))
traceplot(jags.mcmc.heathrow8[,1551:1600], params = c("sigma2_23", "sigma2_24", "Y2", "rmse.Y"))
traceplot(jags.mcmc.heathrow8[,1466:1475], params = c("sigma2_23", "sigma2_24", "Y2", "rmse.Y"))
traceplot(jags.mcmc.heathrow8[,1731:1780], params = c("sigma2_23", "sigma2_24", "Y2", "rmse.Y"))

# for summary statistics
summary(jags.mcmc.heathrow8)
# Iterations = 5001:10000
# Thinning interval = 1
# Number of chains = 2
# Sample size per chain = 5000
```

```
# 1. Empirical mean and standard deviation for each variable, plus standard error of the mean:
#                Mean       SD   Naive SE  Time-series SE
# deviance     9.018e+03 83.41858 0.8341858      39.125147
# Parameters with associated measurements
# rmse.Y[1090]  2.332e-01  0.03945 0.0003945       0.009574
# rmse.Y[1091]  1.698e-01  0.04153 0.0004153       0.007655
# rmse.Y[1092]  3.747e-01  0.04502 0.0004502       0.006194
# rmse.Y[1093]  2.140e-01  0.04976 0.0004976       0.004487
# rmse.Y[1094]  1.394e-01  0.05510 0.0005510       0.004213
# rmse.Y[1095]  2.136e-01  0.06003 0.0006003       0.004596
# rmse.Y[1096]  3.453e-01  0.06405 0.0006405       0.005080
# rmse.Y[1097]  1.166e+00  0.06681 0.0006681       0.005414
# rmse.Y[1098]  1.040e+00  0.06801 0.0006801       0.005334
# rmse.Y[1099]  2.909e-01  0.06807 0.0006807       0.004946
# Parameters with missing data
# rmse.Y[1100]  3.487e-01  0.26172 0.0026172       0.002617
# rmse.Y[1101]  3.484e-01  0.26207 0.0026207       0.002620
# rmse.Y[1102]  3.471e-01  0.26580 0.0026580       0.002687
# rmse.Y[1103]  3.526e-01  0.26477 0.0026477       0.002584
# rmse.Y[1104]  3.473e-01  0.26372 0.0026372       0.002637
# rmse.Y[1105]  3.452e-01  0.26355 0.0026355       0.002727
# rmse.Y[1106]  3.462e-01  0.26300 0.0026300       0.002585
# rmse.Y[1107]  3.470e-01  0.26435 0.0026435       0.002643
# rmse.Y[1108]  3.470e-01  0.26455 0.0026455       0.002583
# rmse.Y[1109]  3.448e-01  0.26386 0.0026386       0.002638

# 2. Quantiles for each variable:
#                2.5%       25%       50%       75%     97.5%
# deviance     8.908e+03  8.951e+03  9.000e+03  9.073e+03  9.204e+03
# Parameters with associated measurements
# rmse.Y[1090]  1.783e-01  2.095e-01  2.246e-01  2.476e-01  3.585e-01
# rmse.Y[1091]  1.086e-01  1.407e-01  1.596e-01  1.951e-01  2.850e-01
# rmse.Y[1092]  3.080e-01  3.381e-01  3.640e-01  4.104e-01  4.776e-01
# rmse.Y[1093]  1.403e-01  1.708e-01  2.022e-01  2.608e-01  3.037e-01
# rmse.Y[1094]  5.471e-02  9.080e-02  1.270e-01  1.938e-01  2.243e-01
# rmse.Y[1095]  1.193e-01  1.611e-01  1.990e-01  2.733e-01  3.039e-01
# rmse.Y[1096]  2.454e-01  2.895e-01  3.265e-01  4.082e-01  4.420e-01
# rmse.Y[1097]  1.062e+00  1.108e+00  1.144e+00  1.233e+00  1.266e+00
# rmse.Y[1098]  9.379e-01  9.802e-01  1.013e+00  1.111e+00  1.139e+00
# rmse.Y[1099]  1.947e-01  2.296e-01  2.663e-01  3.628e-01  3.896e-01
# Parameters with missing data
# rmse.Y[1100]  1.357e-02  1.412e-01  2.977e-01  5.010e-01  9.749e-01
# rmse.Y[1101]  1.532e-02  1.399e-01  2.946e-01  5.014e-01  9.825e-01
# rmse.Y[1102]  1.298e-02  1.382e-01  2.904e-01  4.968e-01  9.909e-01
# rmse.Y[1103]  1.426e-02  1.451e-01  2.987e-01  5.037e-01  9.929e-01
# rmse.Y[1104]  1.403e-02  1.384e-01  2.947e-01  4.999e-01  9.826e-01
# rmse.Y[1105]  1.392e-02  1.356e-01  2.869e-01  4.991e-01  9.893e-01
# rmse.Y[1106]  1.265e-02  1.364e-01  2.935e-01  4.999e-01  9.746e-01
# rmse.Y[1107]  1.411e-02  1.363e-01  2.926e-01  5.033e-01  9.870e-01
# rmse.Y[1108]  1.328e-02  1.396e-01  2.926e-01  4.960e-01  9.863e-01
# rmse.Y[1109]  1.344e-02  1.350e-01  2.892e-01  4.930e-01  9.878e-01


# Produce an MCMC trace of sigma2_23 output
MCMCtrace(jags.mcmc.heathrow8,
        params = c('sigma2_23'),
        type = 'density',
```

```
        ind = TRUE,
        ISB = FALSE,
        pdf = FALSE,
        col_den = c("#aa0078", "#0093af"))

# Produce an MCMC trace of sigma2_24 output
MCMCtrace(jags.mcmc.heathrow8,
        params = c('sigma2_24'),
        type = 'density',
        ind = TRUE,
        ISB = FALSE,
        pdf = FALSE,
        col_den = c("#aa0078", "#0093af"))

# Produce an MCMC trace of rmse.Y output
MCMCtrace(jags.mcmc.heathrow8,
        params = c('rmse.Y'),
        type = 'density',
        ind = TRUE,
        ISB = FALSE,
        pdf = FALSE,
        col_den = c("#aa0078", "#0093af"))

# Find the within-sequence variance (W) and see how close to 0 it is
gelman.heathrow8 <- gelman.diag(jags.mcmc.heathrow8, multivariate = FALSE)
W.heathrow8 <- gelman.heathrow8$W

# Investigate the eigenvalues for W
evals.W.heathrow8 <- eigen(W.heathrow8, only.values = TRUE)$values
min(evals.W.heathrow8)

# Indeed, increasing the tolerance shows that W is indeed positive definite.
matrixNormal::is.positive.definite(W.heathrow8, tol = 1e-18)

gelman.heathrow8
# Potential scale reduction factors:
#          Point est. Upper C.I.
# deviance       1.05       1.06
# sigma2_7       1.02       1.02
# sigma2_8       1.00       1.00
# Y2[1]          4.14       9.09
# Parameters with associated measurements
# Y2[1090]       1.29       2.14
# Y2[1091]       1.14       1.35
# Y2[1092]       1.11       1.16
# Y2[1093]       1.27       2.02
# Y2[1094]       1.86       3.69
# Y2[1095]       2.72       5.68
# Y2[1096]       3.65       7.73
# Y2[1097]       4.54       9.84
# Y2[1098]       5.39      12.12
# Y2[1099]       5.90      13.68
# Parameters with missing data
# Y2[1100]       5.93      14.22
# Y2[1101]       5.77      14.39
# Y2[1102]       5.58      14.51
# Y2[1103]       5.44      14.73
```

```
# Y2[1104]      5.34     14.87
# Y2[1105]      5.31     14.73
# Y2[1106]      5.29     14.11
# Y2[1107]      5.24     13.05
# Y2[1108]      5.16     11.90
# Y2[1109]      5.08     11.14

# Multivariate psrf
# 116

# Create a data.frame from the gelman.diag output to put into a kable table
rhat.heathrow8 <- as.data.frame((gelman.heathrow8)$psrf)
kbl(rhat.heathrow8) %>% kable_styling()

# Create a gelman plot of all the parameters and their densities
gelman.plot(jags.mcmc.heathrow8)


# heathrow data for 2000-2003 and first week of 2004
heathrow4c <- heathrow4[3:1468,]
tail(heathrow4c)

# Plotting the measurements, estimates and rmse for 2000-2003 and first week of 2004 for RW model 1 with rmse.Y
mu.rmse.heathrow9 <- jags.mod.fit.heathrow8$BUGSoutput$mean$rmse.Y
sd.rmse.heathrow9 <- jags.mod.fit.heathrow8$BUGSoutput$sd$rmse.Y

# ```{r}
df.heathrow16 <- data.frame(x9 = heathrow4c$Date, y14 = mu.rmse.heathrow9,
                            lower.rmse.heathrow9 = mu.rmse.heathrow9 - 1.96*sd.rmse.heathrow9,
                            upper.rmse.heathrow9 = mu.rmse.heathrow9 + 1.96*sd.rmse.heathrow9)

ggplot(data = df.heathrow16) +
  geom_point(aes(x = x9, y = y14), colour = 'grey31', size = 2) +
  geom_line(aes(x = x9, y = y14), colour = '#0093af', size = 1) +
  geom_line(aes(x = x9, y = lower.rmse.heathrow9), linetype = "dashed", colour = "#aa0078", size = 1) +
  geom_line(aes(x = x9, y = upper.rmse.heathrow9), linetype = "dashed", colour = "#aa0078", size = 1) +
  xlab('Date') + ylab('Root mean square prediction values of PM10 at Heathrow') +
  ggtitle('PM10 rmse at Heathrow for 2000-2003 and first week of 2004 with RW model 2') +
  theme(axis.title = element_text(size = 14),
        axis.text = element_text(size = 12),
        plot.title = element_text(size = 14))

# When comparing the Y2 estimates with the actual measurements (Heathrow) for RW2, where there are data the
rmse
# values have some variation but always remain above 0.1 and typically below 0.75. The 95% credible interval also
# sometimes has negative lower interval boundary, which is not possible, and a upper interval boundary that
# typically remains below 1. However, where there are no data, the rmse value remains relatively constant
# around a value of 0.02, with a constant credible interval (-0.15, 0.87). This means that when forcasting, the
# forecasted values will tend towards a constant value based on previous estimates and measurements and have an
# expanding credible interval in which the forecasted values would lie.



# We are now going to repeat this analysis for the Haringey site:

## 10. [8 marks] Fit the RW(1) and RW(2) models in JAGS to the Haringey data for 2000 to 2003. Use non-informative
priors.
```

Comment on how well the chains have converged and how well both models fit the data.

```
## Reading in London pollution data
london_pollution <- read.csv("London_Pollution.csv")
london_pollution$Date <- as.Date(london_pollution$Date, format = "%d/%m/%Y")
head(london_pollution)
tail(london_pollution)
london_pollution[350:400,]
london_pollution[50:100,]

## The date and heathrow air pollution data are in the 2nd and 3rd columns, respectively,
## and there are 1827 measurements from 1st January 2000 to 31st December 2004:
haringey <- dplyr::select(london_pollution, Date, Haringey)
head(haringey)
tail(haringey)
count(haringey)
sapply(haringey, class)

# Creating new Day, Month and Year columns from Date column for haringey data
haringey <- haringey %>% separate(Date, into = c('Year', 'Month', 'Day'))
haringey2 <- cbind(haringey, Date = london_pollution$Date)
head(haringey2)
# arrange columns with Date, Day, Month, Year, haringey air pollution
haringey2 <- haringey2 %>% dplyr::select(Date, Year, Month, Day, Haringey)
head(haringey2)
# haringey data for just 2000-2003
haringey3 <- haringey2 %>% dplyr::filter(Year != 2004)
tail(haringey3)
haringey3[1001:1100,]
haringey3[1101:1200,]
haringey3[1201:1300,]



# RW1 model for Haringey data

# Set seed for reproducibility
set.seed(234)
# Set N to be the length of the data
N3 <- length(haringey3$Haringey)
N3
# Mean of haringey data
mean.haringey <- mean(haringey3$Haringey, na.rm = TRUE)
mean.haringey
# Extract haringey column data
Haringey <- haringey3$Haringey
Haringey


# List the data to be used
jags.data.haringey <- list("Haringey", "N3")

# Model
jags.mod.haringey <- function(){
  Z[1] ~ dnorm(0, 0.001)
  for(t in 2:N3){
    Haringey[t] ~ dnorm(Z[t], tau13)            # normal likelihood of haringey data
```

```
   Z[t] ~ dnorm(Z[t-1], tau14)              # normal prediction model
  }
  # priors on measurement error and white noise variances
  tau13 ~ dgamma(0.001, 0.001)              # gamma measurement error precision model
  sigma2_13 <- 1/tau13                      # measurement error variance
  tau14 ~ dgamma(0.001, 0.001)              # gamma estimate error precision model
  sigma2_14 <- 1/tau14                      # estimate error variance
}

# Specify initial values
inits.haringey1 <- list("Z[1]" = 22, "is.na(Haringey)" = mean.haringey)
inits.haringey2 <- list("Z[1]" = 20, "is.na(Haringey)" = mean.haringey)
jags.inits.haringey <- list(inits.haringey1, inits.haringey2)

# Monitor the parameters to be used for the prediction
jags.param.haringey <- c("sigma2_13","sigma2_14","Z")

# Fitting the new model
jags.mod.fit.haringey <- jags(data = jags.data.haringey, inits = jags.inits.haringey,
                      parameters.to.save = jags.param.haringey, n.chains = 2,
                      n.iter = 10000, n.burnin = 5000, n.thin = 1,
                      model.file = jags.mod.haringey)

# Get point and interval estimates
print(jags.mod.fit.haringey)
# Inference for Bugs model at
"/var/folders/pc/hksslngn6_56y6dc8464zthm0000gn/T//Rtmp2LnJkG/model2b92490c5aa.txt", fit using jags,
# 2 chains, each with 10000 iterations (first 5000 discarded)
# n.sims = 10000 iterations saved
#          mu.vect sd.vect    2.5%     25%     50%     75%   97.5%  Rhat n.eff
# Z[1]      10.421   5.898  -1.457   6.418  10.417  14.418  21.917 1.001  7200
# Z[2]      10.770   3.246   4.422   8.588  10.821  12.889  17.116 1.001  4200
# Z[3]      10.136   2.934   4.472   8.127  10.143  12.125  15.919 1.001 10000
# Z[4]      14.244   2.820   8.740  12.349  14.192  16.150  19.857 1.001 10000
# Z[5]      18.512   2.860  12.855  16.580  18.535  20.454  24.038 1.001 10000
# Z[6]      20.137   2.895  14.404  18.217  20.154  22.103  25.697 1.001 10000
# Z[7]      21.901   2.906  16.116  19.989  21.928  23.837  27.689 1.001 10000
# Z[8]      21.320   2.902  15.671  19.395  21.347  23.256  27.036 1.001  7200
# Z[9]      24.266   2.941  18.438  22.364  24.301  26.199  30.024 1.002 10000
# Z[10]     28.646   3.160  22.297  26.715  28.756  30.717  34.448 1.001 10000
# For each parameter, n.eff is a crude measure of effective sample size,
# and Rhat is the potential scale reduction factor (at convergence, Rhat=1).

# DIC info (using the rule, pD = var(deviance)/2)
# pD = 34897.1 and DIC = 41485.7
# DIC is an estimate of expected predictive error (lower deviance is better).



# plot density manually
sim.values.haringey <- jags.mod.fit.haringey$BUGSoutput$sims.list

df.haringey <- data.frame(sigma2_13 = sim.values.haringey$sigma2_13,
                    sigma2_14 = sim.values.haringey$sigma2_14)

haringey_sigma2_13 <- ggplot(data = df.haringey, aes(x = sigma2_13)) +
  geom_density(colour = '#0093af', fill = '#0093af', alpha = 0.3) +
```

```
  theme(axis.title = element_blank(),
        axis.text = element_text(size = 12),
        plot.title = element_text(size = 14)) +
  geom_vline(xintercept = 0, linetype = "dashed",
             colour = "#aa0078", size = 1) +
  ggtitle('Posterior density of sigma2_13 - measurement variance')
haringey_sigma2_13

haringey_sigma2_14 <- ggplot(data = df.haringey, aes(x = sigma2_14)) +
  geom_density(colour = '#0093af', fill = '#0093af', alpha = 0.3) +
  theme(axis.title = element_blank(),
        axis.text = element_text(size = 12),
        plot.title = element_text(size = 14)) +
  geom_vline(xintercept = 0, linetype = "dashed",
             colour = "#aa0078", size = 1) +
  ggtitle('Posterior density of sigma2_14 - estimate variance')
haringey_sigma2_14


# Create an MCMC object from the output of the haringey model
jags.mcmc.haringey <- as.mcmc(jags.mod.fit.haringey)
# Graphical parameters
par(mar= c(2,4,4,2), cex=1.0)
layout(matrix(c(1,1,2,2),2,2,byrow=TRUE))
# for a traceplot, producing for groups of 100 traceplots
traceplot(jags.mcmc.haringey[,1401:1461], params = c("sigma2_13", "sigma2_14", "Z"))
traceplot(jags.mcmc.haringey[,1:50], params = c("sigma2_13", "sigma2_14", "Z"))
traceplot(jags.mcmc.haringey[,201:250], params = c("sigma2_13", "sigma2_14", "Z"))

# Trace plots were generated from jags.mcmc.haringey for each parameter (sigma2_13, sigma2_14,
# deviance, Z[1] and selected examples of Y that covered both where measurements are known and
# where they are missing). The trace plots for deviance, sigma2_13 and sigma2_14 are very close
# to converging for each chain however there is a small amount deviation from each other.
# For the Z estimates, the amount of convergence relies greatly on whether the parameter estimate
# has associated measurements in the haringey dataset or are missing values. The chains converge
# if the Z estimate has an associated haringey measurement and do not converge if there are missing
# values. However, the amount that a Z estimate does not converge becomes greater if the estimate
# is further from a Z estimate that has an associated measurement.

# In this way, Z estimates 975-1210 are missing data, but the convergence of Z estimate 1025 is
# worse than Z estimate 985, which is worse than Z estimate 975. Conversely, the convergence of
# Z estimate 1261 (with ass. measurement) is better than Z estimate 1219 (with ass. measurement),
# which is better than Z estimate 1211 (missing data), which again is better than Z estimate 1210
# (missing data). This shows that as each estimate is based on the one previous, there is a "memory"
# of previous Z estimates that reduces the convergence where estimates "go further into" blocks
# missing data and improves the convergence as the estimates become associated with measurements once more.


# for summary statistics
summary(jags.mcmc.haringey)
summary(jags.mcmc.haringey[,201:250])
# Iterations = 5001:10000
# Thinning interval = 1
# Number of chains = 2
# Sample size per chain = 5000

# 1. Empirical mean and standard deviation for each variable, plus standard error of the mean:
```

```
#              Mean       SD  Naive SE Time-series SE
# deviance  6588.617 264.175  2.64175      21.73989
# sigma2_13   15.715   7.253  0.07253       0.55413
# sigma2_14   24.750   3.954  0.03954       0.35677
# Z[1]        10.421   5.898  0.05898       0.07955
# Parameters with missing data
# Z[1200]     22.106  14.661  0.14661       1.98044
# Z[1201]     22.094  14.333  0.14333       1.75293
# Z[1202]     22.099  13.906  0.13906       1.57161
# Z[1203]     22.140  13.355  0.13355       1.42389
# Z[1204]     22.185  12.696  0.12696       1.22771
# Z[1205]     22.185  11.883  0.11883       0.92774
# Z[1206]     22.233  11.066  0.11066       0.79552
# Z[1207]     22.312  10.049  0.10049       0.66077
# Z[1208]     22.305   9.021  0.09021       0.56751
# Z[1209]     22.285   7.699  0.07699       0.40221
# Parameters with associated measurements
# Z[1210]     22.329   6.096  0.06096       0.28403
# Z[1211]     22.463   3.573  0.03573       0.16824
# Z[1212]     16.997   2.994  0.02994       0.06591
# Z[1213]     17.167   2.938  0.02938       0.06037
# Z[1214]     13.816   2.876  0.02876       0.03676
# Z[1215]     14.883   2.871  0.02871       0.04192
# Z[1216]     14.967   2.861  0.02861       0.03860
# Z[1217]     14.418   2.854  0.02854       0.03611
# Z[1218]     13.456   2.869  0.02869       0.03433
# Z[1219]     14.121   2.865  0.02865       0.03623


# 2. Quantiles for each variable:
#              2.5%       25%      50%      75%    97.5%
# deviance  6193.0949 6459.92395 6579.181 6691.248 6958.68
# sigma2_13   10.7146   13.53034   15.022   16.524   20.98
# sigma2_14   17.2720   22.81586   24.919   27.064   31.27
# Z[1]        -1.4571    6.41790   10.417   14.418   21.92
# Parameters with missing data
# Z[1200]    -6.2102    12.172    21.87    32.41    50.57
# Z[1201]    -5.3159    12.286    22.02    32.20    50.03
# Z[1202]    -4.0144    12.448    21.97    31.41    49.65
# Z[1203]    -2.8004    12.923    21.81    30.98    49.31
# Z[1204]    -1.6920    13.610    21.96    30.55    47.94
# Z[1205]    -0.5831    14.289    21.85    29.88    46.83
# Z[1206]     1.2764    14.637    22.04    29.60    44.65
# Z[1207]     2.7747    15.448    22.23    29.02    42.21
# Z[1208]     4.4151    16.300    22.43    28.33    39.78
# Z[1209]     6.7204    17.175    22.23    27.45    37.28
# Parameters with associated measurements
# Z[1210]     9.6901    18.393    22.44    26.44    34.00
# Z[1211]    15.1674    20.365    22.61    24.86    28.89
# Z[1212]    11.0602    15.014    17.02    19.02    22.73
# Z[1213]    11.2624    15.223    17.19    19.15    22.91
# Z[1214]     8.2030    11.851    13.80    15.77    19.54
# Z[1215]     9.2829    12.945    14.90    16.82    20.55
# Z[1216]     9.3732    13.033    14.95    16.89    20.65
# Z[1217]     8.7231    12.506    14.43    16.37    19.95
# Z[1218]     7.7814    11.516    13.45    15.42    19.09
# Z[1219]     8.5325    12.195    14.08    16.05    19.77
```

```
# Produce an MCMC trace of sigma2_13 output
MCMCtrace(jags.mcmc.haringey,
        params = c('sigma2_13'),
        type = 'density',
        ind = TRUE,
        ISB = FALSE,
        pdf = FALSE,
        col_den = c("#aa0078", "#0093af"))

# Produce an MCMC trace of sigma2_14 output
MCMCtrace(jags.mcmc.haringey,
        params = c('sigma2_14'),
        type = 'density',
        ind = TRUE,
        ISB = FALSE,
        pdf = FALSE,
        col_den = c("#aa0078", "#0093af"))

# Produce an MCMC trace of Z output
MCMCtrace(jags.mcmc.haringey,
        params = c('Z'),
        type = 'density',
        ind = TRUE,
        ISB = FALSE,
        pdf = FALSE,
        col_den = c("#aa0078", "#0093af"))

gelman.haringey <- gelman.diag(jags.mcmc.haringey)
gelman.haringey
# Potential scale reduction factors:
#           Point est. Upper C.I.
# deviance        1.00       1.00
# sigma2_13       1.01       1.01
# sigma2_14       1.00       1.00
# Z[1]            1.00       1.00
# Parameters with missing data
# Z[1200]         1.00       1.00
# Z[1201]         1.00       1.00
# Z[1202]         1.00       1.00
# Z[1203]         1.00       1.00
# Z[1204]         1.00       1.01
# Z[1205]         1.00       1.01
# Z[1206]         1.00       1.00
# Z[1207]         1.00       1.00
# Z[1208]         1.00       1.00
# Z[1209]         1.00       1.00
# Parameters with associated measurements
# Z[1210]         1.00       1.00
# Z[1211]         1.00       1.00
# Z[1212]         1.00       1.00
# Z[1213]         1.00       1.00
# Z[1214]         1.00       1.00
# Z[1215]         1.00       1.00
# Z[1216]         1.00       1.00
# Z[1217]         1.00       1.00
```

```
# Z[1218]       1.00      1.00
# Z[1219]       1.00      1.00

# Multivariate psrf
# 2.23

# Create a data.frame from the gelman.diag output to put into a kable table
rhat.haringey <- as.data.frame((gelman.haringey)$psrf)
kbl(rhat.haringey) %>% kable_styling()

# Create a gelman plot of all the parameters and their densities
gelman.plot(jags.mcmc.haringey)


# Below is the code to plot the Haringey measurements (NA values removed) against the Z estimates (including
where
# missing data has been estimated) and the associated upper and lower 95% credible intervals for the Z estimates.
# Typically, the measurements and the Z estimates are similar to each other and the Heathrow measurements are
within
# the 95% credible intervals. Further, these credible intervals are tightly constrained the the Z estimates.
# However, where there are missing data, the Z estimates deviate from the mean value of the Haringey
measurements
# and the credible interval expands greatly, so much so that the lower interval boundary is in negative numbers.
# Of course, such numbers have no meaning as all measurements must be >0 (i.e. positive). Once there are
measurements
# to constrain the Z estimates, these estimates track closely to the measurements and the credible intervals move
# to be close to the estimates once more. Consequently, this RW1 model fits the data well.

# ```{r}
mu.haringey <- jags.mod.fit.haringey$BUGSoutput$mean$Z
sd.haringey <- jags.mod.fit.haringey$BUGSoutput$sd$Z

# ```{r}
df.haringey2 <- data.frame(x = haringey3$Date, y1 = mu.haringey, y2 = haringey3$Haringey,
                           lower.haringey = mu.haringey - 1.96*sd.haringey,
                           upper.haringey = mu.haringey + 1.96*sd.haringey)

ggplot(data = df.haringey2) +
  geom_point(aes(x = x, y = y1), colour = 'grey31', size = 2) +
  geom_point(aes(x = x, y = y2), colour = 'blue', size = 2) +
  geom_line(aes(x = x, y = y1), colour = '#0093af', size = 1) +
  geom_line(aes(x = x, y = lower.haringey), linetype = "dashed", colour = "#aa0078", size = 1) +
  geom_line(aes(x = x, y = upper.haringey), linetype = "dashed", colour = "#aa0078", size = 1) +
  xlab('Date') + ylab('Z (estimates) and Haringey (measurements) of PM10 at Haringey') +
  ggtitle('PM10 measurements and estimates at Haringey for 2000-2003') +
  theme(axis.title = element_text(size = 14),
        axis.text = element_text(size = 12),
        plot.title = element_text(size = 14))



# RW2 model for Haringey data

# Set seed for reproducibility
set.seed(234)
# Set N to be the length of the data
N1 <- length(haringey3$Haringey)
```

```
N1
# Mean of haringey data
mean.haringey <- mean(haringey3$Haringey, na.rm = TRUE)
mean.haringey
# Extract haringey column data
Haringey <- haringey3$Haringey
Haringey


# List the data to be used
jags.data.haringey2 <- list("Haringey", "N1")

# Model 2
jags.mod.haringey2 <- function(){
  Z[1] ~ dnorm(0, 0.001)
  Z[2] ~ dnorm(0, 0.001)
  for(t in 3:N1){
    Haringey[t] ~ dnorm(Z[t], tau15)            # normal likelihood of haringey data
    Z[t] ~ dnorm((2*Z[t-1] - Z[t-2]), tau16)      # normal prediction model
  }
  # priors on measurement error and white noise variances
  tau15 ~ dgamma(0.001, 0.001)                  # gamma measurement error precision model
  sigma2_15 <- 1/tau15                        # measurement error variance
  tau16 ~ dgamma(0.001, 0.001)                  # gamma estimate error precision model
  sigma2_16 <- 1/tau16                        # estimate error variance
}

# Specify initial values
inits.haringey3 <- list("Z[1]" = 22, "is.na(Haringey)" = mean.haringey)
inits.haringey4 <- list("Z[1]" = 20, "is.na(Haringey)" = mean.haringey)
jags.inits.haringey2 <- list(inits.haringey3, inits.haringey4)

# Monitor the parameters to be used for the prediction
jags.param.haringey2 <- c("sigma2_15","sigma2_16","Z")

# Fitting the new model
jags.mod.fit.haringey2 <- jags(data = jags.data.haringey2, inits = jags.inits.haringey2,
                      parameters.to.save = jags.param.haringey2, n.chains = 2,
                      n.iter = 10000, n.burnin = 5000, n.thin = 1,
                      model.file = jags.mod.haringey2)

# Get point and interval estimates
print(jags.mod.fit.haringey2)
# Inference for Bugs model at
"/var/folders/pc/hksslngn6_56y6dc8464zthm0000gn/T//Rtmp2LnJkG/model2b919fde4a0.txt", fit using jags,
# 2 chains, each with 10000 iterations (first 5000 discarded)
# n.sims = 10000 iterations saved
#          mu.vect sd.vect   2.5%     25%     50%     75%     97.5%   Rhat n.eff
# Z[1]      11.228   2.959   3.773   9.866   11.462   13.016    15.870  1.684    5
# Z[2]      10.983   2.513   4.518   9.861   11.248   12.357    15.074  1.558    6
# Z[3]      10.733   2.146   5.392   9.764   11.028   11.796    14.368  1.267   11
# Z[4]      10.480   1.882   6.097   9.618   10.662   11.597    13.709  1.122   19
# Z[5]      10.222   1.740   6.447   9.403   10.351   11.461    13.062  1.013  130
# Z[6]       9.954   1.724   6.482   8.892   10.238   11.335    12.475  1.032  110
# Z[7]       9.677   1.803   6.316   8.203   10.101   11.087    12.186  1.170   16
# Z[8]       9.396   1.943   5.582   7.693    9.857   10.984    12.153  1.407    7
# Z[9]       9.107   2.119   4.761   7.264    9.671   10.756    12.245  1.675    5
```

```
# Z[10]      8.811  2.318   4.012   6.890   9.488  10.530   12.314  1.933    4

# For each parameter, n.eff is a crude measure of effective sample size,
# and Rhat is the potential scale reduction factor (at convergence, Rhat=1).

# DIC info (using the rule, pD = var(deviance)/2)
# pD = 3260.5 and DIC = 13142.6
# DIC is an estimate of expected predictive error (lower deviance is better).


# Create an MCMC object from the output of the haringey model
jags.mcmc.haringey2 <- as.mcmc(jags.mod.fit.haringey2)
# Graphical parameters
par(mar= c(2,4,4,2), cex=1.0)
layout(matrix(c(1,1,2,2),2,2,byrow=TRUE))
# for a traceplot, producing for groups of 100 traceplots
traceplot(jags.mcmc.haringey2[,1401:1461], params = c("sigma2_15", "sigma2_16", "Z"))
traceplot(jags.mcmc.haringey2[,1:50], params = c("sigma2_15", "sigma2_16", "Z"))
traceplot(jags.mcmc.haringey2[,201:250], params = c("sigma2_15", "sigma2_16", "Z"))

# for summary statistics
summary(jags.mcmc.haringey2)
summary(jags.mcmc.haringey2[,201:250])
# Iterations = 5001:10000
# Thinning interval = 1
# Number of chains = 2
# Sample size per chain = 5000

# 1. Empirical mean and standard deviation for each variable, plus standard error of the mean:
#             Mean       SD  Naive SE  Time-series SE
# deviance   9.882e+03 93.89876 0.9389876      39.355024
# sigma2_15  2.460e+02 21.87206 0.2187206       6.366738
# sigma2_16  4.214e-02  0.08405 0.0008405       0.006177
# Z[1]       1.123e+01  2.95921 0.0295921       0.525142
# Parameters with missing data
# Z[1200]    1.16323    1.0312  0.010312        0.2607
# Z[1201]    1.18493    1.1751  0.011751        0.3225
# Z[1202]    1.23549    1.3646  0.013646        0.3757
# Z[1203]    1.32402    1.5714  0.015714        0.4784
# Z[1204]    1.45487    1.7676  0.017676        0.6201
# Z[1205]    1.62723    1.9363  0.019363        0.6928
# Z[1206]    1.84496    2.0595  0.020595        0.7901
# Z[1207]    2.11060    2.1217  0.021217        0.8757
# Z[1208]    2.42274    2.1121  0.021121        0.8887
# Z[1209]    2.78181    2.0359  0.020359        0.9019
# Parameters with associated measurements
# Z[1210]    3.18585    1.9093  0.019093        0.8975
# Z[1211]    3.62632    1.7534  0.017534        0.8647
# Z[1212]    4.08732    1.6005  0.016005        0.7906
# Z[1213]    4.56000    1.4883  0.014883        0.6971
# Z[1214]    5.03802    1.4389  0.014389        0.6523
# Z[1215]    5.51494    1.4537  0.014537        0.6201
# Z[1216]    5.98101    1.5243  0.015243        0.6261
# Z[1217]    6.42778    1.6381  0.016381        0.6281
# Z[1218]    6.84249    1.7757  0.017757        0.6038
# Z[1219]    7.21988    1.9091  0.019091        0.5979
```

```
# 2. Quantiles for each variable:
#              2.5%       25%       50%       75%     97.5%
# deviance   9.696e+03  9.821e+03  9.888e+03 9940.33447  1.006e+04
# sigma2_15  2.045e+02  2.314e+02  2.464e+02  260.12012  2.890e+02
# sigma2_16  8.781e-03  1.896e-02  2.819e-02    0.03617  1.747e-01
# Z[1]       3.773e+00  9.866e+00  1.146e+01   13.01645  1.587e+01
# Parameters with missing data
# Z[1200]   -1.01484    0.69115    1.16620     1.6061   3.639
# Z[1201]   -1.25779    0.58769    1.23414     1.7574   3.974
# Z[1202]   -1.59977    0.46679    1.39274     1.9909   4.226
# Z[1203]   -2.17964    0.36611    1.64420     2.2828   4.463
# Z[1204]   -2.69005    0.36979    1.91245     2.5934   4.689
# Z[1205]   -2.97034    0.40512    2.17930     2.9067   4.960
# Z[1206]   -2.92878    0.44331    2.45157     3.2009   5.239
# Z[1207]   -2.67374    0.58178    2.71814     3.5201   5.536
# Z[1208]   -2.23431    0.84504    2.99621     3.8697   5.718
# Z[1209]   -1.64996    1.27004    3.30337     4.2098   5.826
# Parameters with associated measurements
# Z[1210]   -0.90016    1.80103    3.64428     4.5575   5.995
# Z[1211]   -0.09291    2.48025    3.99302     4.8777   6.220
# Z[1212]    0.76696    3.15478    4.38117     5.1558   6.671
# Z[1213]    1.54248    3.74448    4.75587     5.4409   7.163
# Z[1214]    2.21692    4.15240    5.17713     5.8415   7.657
# Z[1215]    2.84291    4.36737    5.68810     6.3741   8.155
# Z[1216]    3.20800    4.68544    6.15344     7.0178   8.615
# Z[1217]    2.88379    5.19490    6.51278     7.6767   9.089
# Z[1218]    2.79333    5.69671    6.79156     8.2916   9.522
# Z[1219]    2.80202    6.01148    7.18916     8.8965   9.905


# plot density manually
sim.values.haringey2 <- jags.mod.fit.haringey2$BUGSoutput$sims.list

df.haringey3 <- data.frame(sigma2_15 = sim.values.haringey2$sigma2_15,
                           sigma2_16 = sim.values.haringey2$sigma2_16)

haringey_sigma2_15 <- ggplot(data = df.haringey3, aes(x = sigma2_15)) +
  geom_density(colour = '#0093af', fill = '#0093af', alpha = 0.3) +
  theme(axis.title = element_blank(),
        axis.text = element_text(size = 12),
        plot.title = element_text(size = 14)) +
  geom_vline(xintercept = 0, linetype = "dashed",
             colour = "#aa0078", size = 1) +
  ggtitle('Posterior density of sigma2_15 - measurement variance')
haringey_sigma2_15

haringey_sigma2_16 <- ggplot(data = df.haringey3, aes(x = sigma2_16)) +
  geom_density(colour = '#0093af', fill = '#0093af', alpha = 0.3) +
  theme(axis.title = element_blank(),
        axis.text = element_text(size = 12),
        plot.title = element_text(size = 14)) +
  geom_vline(xintercept = 0, linetype = "dashed",
             colour = "#aa0078", size = 1) +
  ggtitle('Posterior density of sigma2_16 - estimate variance')
haringey_sigma2_16
```

```
# Produce an MCMC trace of sigma2_15 output
MCMCtrace(jags.mcmc.haringey2,
        params = c('sigma2_15'),
        type = 'density',
        ind = TRUE,
        ISB = FALSE,
        pdf = FALSE,
        col_den = c("#aa0078", "#0093af"))

# Produce an MCMC trace of sigma2_16 output
MCMCtrace(jags.mcmc.haringey2,
        params = c('sigma2_16'),
        type = 'density',
        ind = TRUE,
        ISB = FALSE,
        pdf = FALSE,
        col_den = c("#aa0078", "#0093af"))

# Produce an MCMC trace of Z output
MCMCtrace(jags.mcmc.haringey2,
        params = c('Z'),
        type = 'density',
        ind = TRUE,
        ISB = FALSE,
        pdf = FALSE,
        col_den = c("#aa0078", "#0093af"))

gelman.haringey2 <- gelman.diag(jags.mcmc.haringey2)
gelman.haringey2
# Potential scale reduction factors:
#           Point est. Upper C.I.
# deviance       1.73      3.56
# sigma2_15      1.49      2.55
# sigma2_16      1.01      1.04
# Z[1]           1.68      2.95
# Parameters with missing data
# Z[1200]        1.19      1.49
# Z[1201]        1.26      1.96
# Z[1202]        1.47      2.41
# Z[1203]        1.69      3.00
# Z[1204]        1.84      3.70
# Z[1205]        1.92      4.33
# Z[1206]        1.92      4.72
# Z[1207]        1.88      4.86
# Z[1208]        1.81      4.81
# Z[1209]        1.72      4.61
# Parameters with associated measurements
# Z[1210]        1.60      4.25
# Z[1211]        1.47      3.65
# Z[1212]        1.33      2.76
# Z[1213]        1.23      1.73
# Z[1214]        1.16      1.17
# Z[1215]        1.13      1.30
# Z[1216]        1.17      1.59
# Z[1217]        1.31      1.98
# Z[1218]        1.49      2.43
```

```
# Z[1219]       1.66      2.86

# Multivariate psrf
# 112

# Create a data.frame from the gelman.diag output to put into a kable table
rhat.haringey2 <- as.data.frame((gelman.haringey2)$psrf)
kbl(rhat.haringey2) %>% kable_styling()

# Create a gelman plot of all the parameters and their densities
gelman.plot(jags.mcmc.haringey2)


# Plotting the measurements and estimates for 2000-2003 for RW model 2
mu.haringey2 <- jags.mod.fit.haringey2$BUGSoutput$mean$Z
sd.haringey2 <- jags.mod.fit.haringey2$BUGSoutput$sd$Z

# ```{r}
df.haringey4 <- data.frame(x2 = haringey3$Date, y3 = mu.haringey2, y4 = haringey3$Haringey,
                           lower.haringey2 = mu.haringey2 - 1.96*sd.haringey2,
                           upper.haringey2 = mu.haringey2 + 1.96*sd.haringey2)

ggplot(data = df.haringey4) +
  geom_point(aes(x = x2, y = y3), colour = 'grey31', size = 2) +
  geom_point(aes(x = x2, y = y4), colour = 'blue', size = 2) +
  geom_line(aes(x = x2, y = y3), colour = '#0093af', size = 1) +
  geom_line(aes(x = x2, y = lower.haringey2), linetype = "dashed", colour = "#aa0078", size = 1) +
  geom_line(aes(x = x2, y = upper.haringey2), linetype = "dashed", colour = "#aa0078", size = 1) +
  xlab('Date') + ylab('Z (estimates) and Haringey (measurements) of PM10 at Haringey') +
  ggtitle('PM10 measurements and estimates at Haringey for 2000-2003 with RW model 2') +
  theme(axis.title = element_text(size = 14),
        axis.text = element_text(size = 12),
        plot.title = element_text(size = 14))



# With Random Walk 1 (RW1) modelling, the estimates are based on the original measurements, so there is
# something like one degree of "freedom" or separation. With random walk 2 (RW2) modelling, estimates are
# based on on previous estimates, that themselves are based on the measurements. This provides something
# like two degrees of "freedom" or separation, so the chains do not converge as there is less to
# constrain them. This RW2 model fits the data poorly as the model sits below the data and does not track them.




## 11. [10 marks] Now re-run these analyses using informative priors, using what you have learnt from fitting the Heathrow model.
## By comparing the results (e.g. summmaries of the posterior distributions, convergence etc), comment on any effect that using
# different priors has (or has not had) on the results.


## Reading in London pollution data
london_pollution <- read.csv("London_Pollution.csv")
london_pollution$Date <- as.Date(london_pollution$Date, format = "%d/%m/%Y")
head(london_pollution)
tail(london_pollution)
```

```
london_pollution[350:400,]
london_pollution[50:100,]

## The date and heathrow air pollution data are in the 2nd and 3rd columns, respectively,
## and there are 1827 measurements from 1st January 2000 to 31st December 2004:
haringey <- dplyr::select(london_pollution, Date, Haringey)
head(haringey)
tail(haringey)
count(haringey)
sapply(haringey, class)

# Creating new Day, Month and Year columns from Date column for haringey data
haringey <- haringey %>% separate(Date, into = c('Year', 'Month', 'Day'))
haringey2 <- cbind(haringey, Date = london_pollution$Date)
head(haringey2)
# arrange columns with Date, Day, Month, Year, haringey air pollution
haringey2 <- haringey2 %>% dplyr::select(Date, Year, Month, Day, Haringey)
head(haringey2)
# haringey data for just 2000-2003
haringey3 <- haringey2 %>% dplyr::filter(Year != 2004)
tail(haringey3)
haringey3[1001:1100,]
haringey3[1101:1200,]
haringey3[1201:1300,]



# RW1 model for Haringey data

# Set seed for reproducibility
set.seed(234)
# Set N to be the length of the data
N3 <- length(haringey3$Haringey)
N3
# Mean of haringey data
mean.haringey <- mean(haringey3$Haringey, na.rm = TRUE)
mean.haringey
# Extract haringey column data
Haringey <- haringey3$Haringey
Haringey

# From jags.mod.haringey2, sigma2_13 and sigma2_14 can inform what tau17 and tau18 are, repsectively
sigma2_13 <- 15.715
tau17 <- 1/sigma2_13
sigma2_14 <- 24.750
tau18 <- 1/sigma2_14

# List the data to be used
jags.data.haringey3 <- list("Haringey", "N3")

# Model
jags.mod.haringey3 <- function(){
  Z2[1] ~ dnorm(0, 0.001)
  for(t in 2:N3){
    Haringey[t] ~ dnorm(Z2[t], tau17)            # normal likelihood of haringey data
    Z2[t] ~ dnorm(Z2[t-1], tau18)                # normal prediction model
  }
```

```
  # priors on measurement error and white noise variances
  tau17 ~ dnorm(0.064, 0.020)              # normal measurement error precision model
  sigma2_17 <- 1/tau17                     # measurement error variance
  tau18 ~ dnorm(0.040, 0.064)              # normal estimate error precision model
  sigma2_18 <- 1/tau18                     # estimate error variance
}

# Specify initial values
inits.haringey5 <- list("Z2[1]" = 22, "is.na(Haringey)" = mean.haringey, "tau17" = 0.064, "tau18" = 0.040)
inits.haringey6 <- list("Z2[1]" = 20, "is.na(Haringey)" = mean.haringey, "tau17" = 0.064, "tau18" = 0.040)
jags.inits.haringey3 <- list(inits.haringey5, inits.haringey6)

# Monitor the parameters to be used for the prediction
jags.param.haringey3 <- c("sigma2_17","sigma2_18","Z2")

# Fitting the new model
jags.mod.fit.haringey3 <- jags(data = jags.data.haringey3, inits = jags.inits.haringey3,
                    parameters.to.save = jags.param.haringey3, n.chains = 2,
                    n.iter = 10000, n.burnin = 5000, n.thin = 1,
                    model.file = jags.mod.haringey3)

# Get point and interval estimates
print(jags.mod.fit.haringey3)
# Inference for Bugs model at
"/var/folders/pc/hksslngn6_56y6dc8464zthm0000gn/T//RtmpSTiK0N/model24a3c22b237.txt", fit using jags,
# 2 chains, each with 10000 iterations (first 5000 discarded)
# n.sims = 10000 iterations saved
#          mu.vect sd.vect   2.5%    25%    50%    75%  97.5%  Rhat n.eff
# Z2[1]     10.503   5.866  -0.932   6.586  10.581  14.469  21.949 1.001  9100
# Z2[2]     10.771   3.267   4.344   8.549  10.790  12.980  17.169 1.001  7400
# Z2[3]     10.096   2.907   4.486   8.119  10.101  12.013  15.822 1.001 10000
# Z2[4]     14.199   2.858   8.532  12.287  14.224  16.097  19.805 1.002  1400
# Z2[5]     18.581   2.841  13.092  16.642  18.574  20.479  24.241 1.002  2100
# Z2[6]     20.189   2.849  14.576  18.258  20.197  22.106  25.809 1.001 10000
# Z2[7]     21.995   2.846  16.400  20.088  21.994  23.924  27.578 1.002   960
# Z2[8]     21.368   2.869  15.637  19.413  21.372  23.325  27.036 1.001 10000
# Z2[9]     24.263   2.839  18.666  22.346  24.223  26.137  29.898 1.001  7700
# Z2[10]    28.720   2.910  23.030  26.744  28.751  30.713  34.289 1.001  3300
# For each parameter, n.eff is a crude measure of effective sample size,
# and Rhat is the potential scale reduction factor (at convergence, Rhat=1).

# DIC info (using the rule, pD = var(deviance)/2)
# pD = 12792.2 and DIC = 19355.4
# DIC is an estimate of expected predictive error (lower deviance is better).


# plot density manually
sim.values.haringey3 <- jags.mod.fit.haringey3$BUGSoutput$sims.list

df.haringey5 <- data.frame(sigma2_17 = sim.values.haringey3$sigma2_17,
                    sigma2_18 = sim.values.haringey3$sigma2_18)

haringey_sigma2_17 <- ggplot(data = df.haringey5, aes(x = sigma2_17)) +
  geom_density(colour = '#0093af', fill = '#0093af', alpha = 0.3) +
  theme(axis.title = element_blank(),
        axis.text = element_text(size = 12),
        plot.title = element_text(size = 14)) +
```

```
  geom_vline(xintercept = 0, linetype = "dashed",
             colour = "#aa0078", size = 1) +
  ggtitle('Posterior density of sigma2_17 - measurement variance')
haringey_sigma2_17

haringey_sigma2_18 <- ggplot(data = df.haringey5, aes(x = sigma2_18)) +
  geom_density(colour = '#0093af', fill = '#0093af', alpha = 0.3) +
  theme(axis.title = element_blank(),
        axis.text = element_text(size = 12),
        plot.title = element_text(size = 14)) +
  geom_vline(xintercept = 0, linetype = "dashed",
             colour = "#aa0078", size = 1) +
  ggtitle('Posterior density of sigma2_18 - estimate variance')
haringey_sigma2_18


# Create an MCMC object from the output of the haringey model
jags.mcmc.haringey3 <- as.mcmc(jags.mod.fit.haringey3)
# Graphical parameters
par(mar = c(2,4,4,2), cex = 1.0)
layout(matrix(c(1,1,2,2),2,2,byrow=TRUE))
# for a traceplot, producing for groups of 100 traceplots
traceplot(jags.mcmc.haringey3[,1401:1461], params = c("sigma2_17", "sigma2_18", "Z2"))
traceplot(jags.mcmc.haringey3[,1:50], params = c("sigma2_17", "sigma2_18", "Z2"))
traceplot(jags.mcmc.haringey3[,201:250], params = c("sigma2_17", "sigma2_18", "Z2"))


# for summary statistics
summary(jags.mcmc.haringey3[,1:5])
summary(jags.mcmc.haringey3[,201:250])
# Iterations = 5001:10000
# Thinning interval = 1
# Number of chains = 2
# Sample size per chain = 5000

# 1. Empirical mean and standard deviation for each variable, plus standard error of the mean:
#             Mean       SD  Naive SE Time-series SE
# deviance  6563.25 160.738  1.60738      11.56295
# sigma2_17   14.95   2.107  0.02107       0.15290
# sigma2_18   25.00   2.991  0.02991       0.21779
# Z2[1]       10.50   5.866  0.05866       0.08293
# Parameters with missing data
# Z2[1200]    21.12  14.063  0.14063       1.35265
# Z2[1201]    20.97  13.628  0.13628       1.32896
# Z2[1202]    20.90  13.221  0.13221       1.27975
# Z2[1203]    20.94  12.714  0.12714       1.15848
# Z2[1204]    21.09  12.070  0.12070       1.08956
# Z2[1205]    21.20  11.441  0.11441       0.95504
# Z2[1206]    21.33  10.607  0.10607       0.75820
# Z2[1207]    21.51   9.661  0.09661       0.67383
# Z2[1208]    21.62   8.631  0.08631       0.49653
# Z2[1209]    21.80   7.392  0.07392       0.37043
# Parameters with associated measurements
# Z2[1210]    22.14   5.775  0.05775       0.22180
# Z2[1211]    22.57   3.239  0.03239       0.07884
# Z2[1212]    17.14   2.878  0.02878       0.03555
# Z2[1213]    17.35   2.858  0.02858       0.03336
```

```
# Z2[1214]    13.89   2.838  0.02838       0.03592
# Z2[1215]    14.91   2.822  0.02822       0.03338
# Z2[1216]    14.98   2.831  0.02831       0.03293
# Z2[1217]    14.33   2.824  0.02824       0.03259
# Z2[1218]    13.36   2.836  0.02836       0.03373
# Z2[1219]    14.13   2.843  0.02843       0.03643

# 2. Quantiles for each variable:
#             2.5%      25%     50%     75%   97.5%
# deviance  6232.4509 6456.792 6567.92 6675.83 6865.66
# sigma2_17   11.0616   13.461   14.88   16.33   19.39
# sigma2_18   19.2650   22.892   24.96   27.12   30.96
# Z2[1]      -0.9324    6.586   10.58   14.47   21.95
# Parameters with associated measurements
# Z2[1200]    -7.2328   11.739   21.39   30.57   48.40
# Z2[1201]    -6.5466   12.035   21.10   30.02   47.45
# Z2[1202]    -6.2826   12.389   20.93   29.49   47.26
# Z2[1203]    -4.7892   12.566   20.97   29.22   46.40
# Z2[1204]    -2.7204   13.016   21.08   29.12   45.00
# Z2[1205]    -1.2165   13.523   21.27   28.82   44.26
# Z2[1206]     0.5347   14.195   21.27   28.40   42.52
# Z2[1207]     2.4646   14.941   21.53   28.05   40.72
# Z2[1208]     4.9693   15.779   21.60   27.40   38.81
# Z2[1209]     7.0770   16.871   21.77   26.65   36.73
# Parameters with missing data
# Z2[1210]    10.5828   18.299   22.18   25.95   33.58
# Z2[1211]    16.1810   20.387   22.60   24.76   28.85
# Z2[1212]    11.5775   15.164   17.10   19.07   22.87
# Z2[1213]    11.7345   15.420   17.37   19.29   22.98
# Z2[1214]     8.4040   11.976   13.86   15.81   19.54
# Z2[1215]     9.3215   13.040   14.91   16.78   20.39
# Z2[1216]     9.3315   13.115   15.03   16.84   20.63
# Z2[1217]     8.7868   12.430   14.33   16.22   19.89
# Z2[1218]     7.8628   11.482   13.34   15.24   19.00
# Z2[1219]     8.5835   12.210   14.15   16.01   19.77


# Produce an MCMC trace of sigma2_17 output
MCMCtrace(jags.mcmc.haringey3,
        params = c('sigma2_17'),
        type = 'density',
        ind = TRUE,
        ISB = FALSE,
        pdf = FALSE,
        col_den = c("#aa0078", "#0093af"))

# Produce an MCMC trace of sigma2_18 output
MCMCtrace(jags.mcmc.haringey3,
        params = c('sigma2_18'),
        type = 'density',
        ind = TRUE,
        ISB = FALSE,
        pdf = FALSE,
        col_den = c("#aa0078", "#0093af"))

# Produce an MCMC trace of Z2 output
MCMCtrace(jags.mcmc.haringey3,
```

```
        params = c('Z2'),
        type = 'density',
        ind = TRUE,
        ISB = FALSE,
        pdf = FALSE,
        col_den = c("#aa0078", "#0093af"))

gelman.haringey3 <- gelman.diag(jags.mcmc.haringey3)
gelman.haringey3
# Potential scale reduction factors:
#          Point est. Upper C.I.
# deviance       1.01      1.01
# sigma2_17      1.00      1.00
# sigma2_18      1.01      1.01
# Z2[1]          1.00      1.00
# Parameters with missing data
# Z2[1200]       1.04      1.15
# Z2[1201]       1.04      1.16
# Z2[1202]       1.05      1.18
# Z2[1203]       1.05      1.21
# Z2[1204]       1.06      1.22
# Z2[1205]       1.06      1.25
# Z2[1206]       1.06      1.25
# Z2[1207]       1.05      1.22
# Z2[1208]       1.05      1.19
# Z2[1209]       1.04      1.14
# Parameters with associated measurements
# Z2[1210]       1.02      1.08
# Z2[1211]       1.00      1.02
# Z2[1212]       1.00      1.00
# Z2[1213]       1.00      1.00
# Z2[1214]       1.00      1.00
# Z2[1215]       1.00      1.00
# Z2[1216]       1.00      1.00
# Z2[1217]       1.00      1.00
# Z2[1218]       1.00      1.00
# Z2[1219]       1.00      1.00

# Multivariate psrf
# 1.82

# Create a data.frame from the gelman.diag output to put into a kable table
rhat.haringey3 <- as.data.frame((gelman.haringey3)$psrf)
kbl(rhat.haringey3) %>% kable_styling()


# Below is the code to plot the Haringey measurements (NA values removed) against the Z2 estimates (including
where
# missing data has been estimated) and the associated upper and lower 95% credible intervals for the Z2 estimates.
# Typically, the measurements and the Z2 estimates are similar to each other and the Heathrow measurements are
within
# the 95% credible intervals. Further, these credible intervals are tightly constrained the the Z2 estimates.
# However, where there are missing data, the Z2 estimates deviate from the mean value of the Haringey
measurements
# and the credible interval expands greatly, so much so that the lower interval boundary is in negative numbers.
# Of course, such numbers have no meaning as all measurements must be >0 (i.e. positive). Once there are
measurements
```

```
# to constrain the Z2 estimates, these estimates track closely to the measurements and the credible intervals move
# to be close to the estimates once more. Consequently, this RW1 model fits the data well.

# ```{r}
mu.haringey3 <- jags.mod.fit.haringey3$BUGSoutput$mean$Z2
sd.haringey3 <- jags.mod.fit.haringey3$BUGSoutput$sd$Z2

# ```{r}
df.haringey6 <- data.frame(x3 = haringey3$Date, y5 = mu.haringey3, y6 = haringey3$Haringey,
                           lower.haringey3 = mu.haringey3 - 1.96*sd.haringey3,
                           upper.haringey3 = mu.haringey3 + 1.96*sd.haringey3)

ggplot(data = df.haringey6) +
  geom_point(aes(x = x3, y = y5), colour = 'grey31', size = 2) +
  geom_point(aes(x = x3, y = y6), colour = 'blue', size = 2) +
  geom_line(aes(x = x3, y = y5), colour = '#0093af', size = 1) +
  geom_line(aes(x = x3, y = lower.haringey3), linetype = "dashed", colour = "#aa0078", size = 1) +
  geom_line(aes(x = x3, y = upper.haringey3), linetype = "dashed", colour = "#aa0078", size = 1) +
  xlab('Date') + ylab('Z2 (estimates) and Haringey (measurements) of PM10 at Haringey') +
  ggtitle('PM10 measurements and estimates at Haringey for 2000-2003 for RW1 using informative priors from
Heathrow modelling') +
  theme(axis.title = element_text(size = 14),
        axis.text = element_text(size = 12),
        plot.title = element_text(size = 14))

# There is little or no improvement in which estimates converged for jags.mod.fit.haringey (using
# non-informative priors) and jags.mod.fit.haringey3 (using informative priors), which both employ
# RW1 models. However, jags.mod.fit.haringey3 has a lower DIC value (= 19355) than for
# jags.mod.fit.haringey (= 41485). Details of the output from jags.mod.fit.haringey3 are given below:

# Trace plots were generated from jags.mcmc.haringey3 for each parameter (sigma2_17, sigma2_18,
# deviance, Z2[1] and selected examples of Z2 that covered both where measurements are known and
# where they are missing). The trace plots for deviance, sigma2_17 and sigma2_18 are very close
# to converging for each chain however there is a small amount deviation from each other.
# For the Z2 estimates, the amount of convergence relies greatly on whether the parameter estimate
# has associated measurements in the haringey dataset or are missing values. The chains converge
# if the Z2 estimate has an associated haringey measurement and do not converge if there are missing
# values. However, the amount that a Z2 estimate does not converge becomes greater if the estimate
# is further from a Z2 estimate that has an associated measurement.

# In this way, Z2 estimates 975-1210 are missing data, but the convergence of Z2 estimate 1025 is
# worse than Z2 estimate 985, which is worse than Z2 estimate 975. Conversely, the convergence of
# Z2 estimate 1261 (with ass. measurement) is better than Z2 estimate 1219 (with ass. measurement),
# which is better than Z2 estimate 1211 (missing data), which again is better than Z2 estimate 1210
# (missing data). This shows that as each estimate is based on the one previous, there is a "memory"
# of previous Z2 estimates that reduces the convergence where estimates "go further into" blocks
# missing data and improves the convergence as the estimates become associated with measurements once more.




# RW2 model for Haringey data

# Set seed for reproducibility
set.seed(234)
# Set N to be the length of the data
N1 <- length(haringey3$Haringey)
```

```
N1
# Mean of haringey data
mean.haringey <- mean(haringey3$Haringey, na.rm = TRUE)
mean.haringey
# Extract haringey column data
Haringey <- haringey3$Haringey
Haringey

# From jags.mod.haringey2, sigma2_15 and sigma2_16 can inform what tau19 and tau20 are, repsectively
sigma2_15 <- 246
tau19 <- 1/sigma2_15
sigma2_16 <- 421.6
tau20 <- 1/sigma2_16

# List the data to be used
jags.data.haringey4 <- list("Haringey", "N1")

# Model 2
jags.mod.haringey4 <- function(){
  Z2[1] ~ dnorm(0, 0.001)
  Z2[2] ~ dnorm(0, 0.001)
  for(t in 3:N1){
    Haringey[t] ~ dnorm(Z2[t], tau19)             # normal likelihood of haringey data
    Z2[t] ~ dnorm((2*Z2[t-1] - Z2[t-2]), tau20)    # normal prediction model
  }
  # informative priors on measurement error and white noise variances
  tau19 ~ dnorm(0.004, 0.002)                  # normal measurement error precision model
  sigma2_19 <- 1/tau19                         # measurement error variance
  tau20 ~ dnorm(0.002, 156.25)                 # normal estimate error precision model
  sigma2_20 <- 1/tau20                         # estimate error variance
}

# Specify initial values
inits.haringey7 <- list("Z2[1]" = 22, "is.na(Haringey)" = mean.haringey, "tau19" = 0.004, "tau20" = 0.002)
inits.haringey8 <- list("Z2[1]" = 20, "is.na(Haringey)" = mean.haringey, "tau19" = 0.004, "tau20" = 0.002)
jags.inits.haringey4 <- list(inits.haringey7, inits.haringey8)

# Monitor the parameters to be used for the prediction
jags.param.haringey4 <- c("sigma2_19", "sigma2_20", "Z2")

# Fitting the new model
jags.mod.fit.haringey4 <- jags(data = jags.data.haringey4, inits = jags.inits.haringey4,
                      parameters.to.save = jags.param.haringey4, n.chains = 2,
                      n.iter = 10000, n.burnin = 5000, n.thin = 1,
                      model.file = jags.mod.haringey4)

# Get point and interval estimates
print(jags.mod.fit.haringey4)
# Inference for Bugs model at
"/var/folders/pc/hksslngn6_56y6dc8464zthm0000gn/T//RtmpSTiK0N/model24a9e257b9.txt", fit using jags,
# 2 chains, each with 10000 iterations (first 5000 discarded)
# n.sims = 10000 iterations saved
#          mu.vect sd.vect   2.5%     25%     50%     75%   97.5%   Rhat  n.eff
# Z2[1]      3.541   8.225  -12.297   -2.227   3.313   9.289   19.551  1.007   260
# Z2[2]      7.005   5.840   -4.108   2.859   6.884   11.111   18.334  1.009   190
# Z2[3]     10.434   4.021   2.648   7.573   10.438   13.259   18.169  1.011   150
# Z2[4]     13.841   2.957   8.092   11.769   13.854   15.806   19.539  1.010   170
```

```
# Z2[5]      16.997   2.546   12.084   15.296   16.986   18.646   22.154  1.006   300
# Z2[6]      19.620   2.491   14.674   17.952   19.591   21.286   24.637  1.006  1000
# Z2[7]      21.626   2.508   16.693   19.966   21.660   23.313   26.435  1.007  2300
# Z2[8]      22.978   2.460   18.125   21.345   23.008   24.641   27.630  1.006  1000
# Z2[9]      23.830   2.380   19.084   22.238   23.831   25.479   28.304  1.005   410
# Z2[10]     23.965   2.364   19.323   22.350   23.983   25.606   28.424  1.005   380

# For each parameter, n.eff is a crude measure of effective sample size,
# and Rhat is the potential scale reduction factor (at convergence, Rhat=1).

# DIC info (using the rule, pD = var(deviance)/2)
# pD = 820.4 and DIC = 8300.2
# DIC is an estimate of expected predictive error (lower deviance is better).


# Create an MCMC object from the output of the haringey model
jags.mcmc.haringey4 <- as.mcmc(jags.mod.fit.haringey4)
# Graphical parameters
par(mar = c(2,4,4,2), cex = 1.0)
layout(matrix(c(1,1,2,2),2,2,byrow=TRUE))
# for a traceplot, producing for groups of 100 traceplots
traceplot(jags.mcmc.haringey4[,1401:1461], params = c("sigma2_19", "sigma2_20", "Z2"))
traceplot(jags.mcmc.haringey4[,1:50], params = c("sigma2_19", "sigma2_20", "Z2"))
traceplot(jags.mcmc.haringey4[,201:250], params = c("sigma2_19", "sigma2_20", "Z2"))

# for summary statistics
summary(jags.mcmc.haringey4)
summary(jags.mcmc.haringey4[,201:250])
# Iterations = 5001:10000
# Thinning interval = 1
# Number of chains = 2
# Sample size per chain = 5000

# 1. Empirical mean and standard deviation for each variable, plus standard error of the mean:
#              Mean      SD  Naive SE  Time-series SE
# deviance  7479.774  40.5524 0.405524       3.44180
# sigma2_19   32.266   1.7006 0.017006       0.10647
# sigma2_20    2.645   0.3115 0.003115       0.02365
# Z2[1]        3.541   8.2245 0.082245       0.49319
# Parameters with missing data
# Z2[1200]     3.1782  9.719  0.09719        2.4870
# Z2[1201]     5.6866  9.212  0.09212        2.5340
# Z2[1202]     8.2279  8.893  0.08893        2.5011
# Z2[1203]    10.6647  8.632  0.08632        2.3311
# Z2[1204]    12.8697  8.338  0.08338        2.1097
# Z2[1205]    14.7988  7.933  0.07933        1.8530
# Z2[1206]    16.4295  7.374  0.07374        1.5546
# Z2[1207]    17.7580  6.676  0.06676        1.2800
# Z2[1208]    18.7439  5.848  0.05848        0.9588
# Z2[1209]    19.3266  4.927  0.04927        0.6982
# Parameters with associated measurements
# Z2[1210]    19.4777  3.971  0.03971        0.4576
# Z2[1211]    19.0885  3.127  0.03127        0.2731
# Z2[1212]    18.0697  2.672  0.02672        0.1992
# Z2[1213]    16.9021  2.568  0.02568        0.1766
# Z2[1214]    15.7237  2.575  0.02575        0.1753
# Z2[1215]    14.8515  2.577  0.02577        0.1711
```

```
# Z2[1216]    14.2331  2.613  0.02613        0.1842
# Z2[1217]    13.8434  2.624  0.02624        0.1880
# Z2[1218]    13.7167  2.581  0.02581        0.1802
# Z2[1219]    13.9165  2.500  0.02500        0.1720

# 2. Quantiles for each variable:
#              2.5%      25%      50%      75%    97.5%
# deviance  7401.396 7452.668 7479.076 7504.643 7563.644
# sigma2_19   29.169   31.089   32.224   33.346   35.818
# sigma2_20    2.098    2.429    2.621    2.833    3.351
# Z2[1]      -12.297   -2.227    3.313    9.289   19.551
# Parameters with missing data
# Z2[1200]  -15.1737   -3.707    3.7970  10.9478  19.5988
# Z2[1201]  -10.8064   -1.254    5.6233  12.9174  22.3750
# Z2[1202]   -6.9077    1.527    7.6365  14.6163  25.4757
# Z2[1203]   -3.4848    3.986    9.9243  16.3887  27.8703
# Z2[1204]   -0.6564    6.371   12.1052  18.2752  30.1391
# Z2[1205]    1.2186    8.811   14.0286  19.6388  31.5486
# Z2[1206]    2.9658   11.127   15.7312  20.7332  32.0204
# Z2[1207]    5.0420   12.984   17.3520  21.6918  31.9918
# Z2[1208]    7.3862   14.731   18.3183  22.4386  31.2796
# Z2[1209]    9.5976   16.125   19.0943  22.3935  29.5282
# Parameters with associated measurements
# Z2[1210]   11.4321   16.942   19.3682  22.0143  27.4445
# Z2[1211]   12.6820   17.112   19.1388  21.1289  25.1248
# Z2[1212]   12.5582   16.332   18.1416  19.8957  23.2297
# Z2[1213]   11.7750   15.151   16.9356  18.6592  21.8880
# Z2[1214]   10.9467   13.911   15.6547  17.4482  20.8205
# Z2[1215]    9.9621   13.056   14.7672  16.6521  20.0088
# Z2[1216]    9.1832   12.396   14.1926  16.0361  19.3462
# Z2[1217]    8.7670   12.038   13.8545  15.6706  18.9519
# Z2[1218]    8.5650   11.953   13.6972  15.4466  18.8006
# Z2[1219]    9.1018   12.239   13.8750  15.5679  18.9476

gelman.haringey4 <- gelman.diag(jags.mcmc.haringey4)
gelman.haringey4
# Potential scale reduction factors:
#             Point est. Upper C.I.
# deviance          1.02       1.04
# sigma2_19         1.01       1.02
# sigma2_20         1.01       1.02
# Z2[1]             1.01       1.03
# Parameters with missing data
# Z2[1200]          1.39       2.27
# Z2[1201]          1.15       1.53
# Z2[1202]          1.05       1.13
# Z2[1203]          1.02       1.02
# Z2[1204]          1.03       1.08
# Z2[1205]          1.05       1.19
# Z2[1206]          1.06       1.25
# Z2[1207]          1.07       1.27
# Z2[1208]          1.07       1.27
# Z2[1209]          1.07       1.26
# Parameters with associated measurements
# Z2[1210]          1.06       1.23
# Z2[1211]          1.04       1.16
# Z2[1212]          1.01       1.06
```

```
# Z2[1213]      1.00     1.01
# Z2[1214]      1.00     1.01
# Z2[1215]      1.00     1.01
# Z2[1216]      1.00     1.02
# Z2[1217]      1.01     1.02
# Z2[1218]      1.01     1.03
# Z2[1219]      1.01     1.04

# Multivariate psrf
# 47.9

# Create a data.frame from the gelman.diag output to put into a kable table
rhat.haringey4 <- as.data.frame((gelman.haringey4)$psrf)
kbl(rhat.haringey4) %>% kable_styling()


# plot density manually
sim.values.haringey4 <- jags.mod.fit.haringey4$BUGSoutput$sims.list

df.haringey7 <- data.frame(sigma2_19 = sim.values.haringey4$sigma2_19,
                           sigma2_20 = sim.values.haringey4$sigma2_20)

haringey_sigma2_19 <- ggplot(data = df.haringey7, aes(x = sigma2_19)) +
  geom_density(colour = '#0093af', fill = '#0093af', alpha = 0.3) +
  theme(axis.title = element_blank(),
        axis.text = element_text(size = 12),
        plot.title = element_text(size = 14)) +
  geom_vline(xintercept = 0, linetype = "dashed",
             colour = "#aa0078", size = 1) +
  ggtitle('Posterior density of sigma2_19 - measurement variance')
haringey_sigma2_19

haringey_sigma2_20 <- ggplot(data = df.haringey7, aes(x = sigma2_20)) +
  geom_density(colour = '#0093af', fill = '#0093af', alpha = 0.3) +
  theme(axis.title = element_blank(),
        axis.text = element_text(size = 12),
        plot.title = element_text(size = 14)) +
  geom_vline(xintercept = 0, linetype = "dashed",
             colour = "#aa0078", size = 1) +
  ggtitle('Posterior density of sigma2_20 - estimate variance')
haringey_sigma2_20



# Produce an MCMC trace of sigma2_19 output
MCMCtrace(jags.mcmc.haringey4,
          params = c('sigma2_19'),
          type = 'density',
          ind = TRUE,
          ISB = FALSE,
          pdf = FALSE,
          col_den = c("#aa0078", "#0093af"))

# Produce an MCMC trace of sigma2_20 output
MCMCtrace(jags.mcmc.haringey4,
          params = c('sigma2_20'),
          type = 'density',
```

```
        ind = TRUE,
        ISB = FALSE,
        pdf = FALSE,
        col_den = c("#aa0078", "#0093af"))

# Produce an MCMC trace of Z2 output
MCMCtrace(jags.mcmc.haringey4,
        params = c('Z2'),
        type = 'density',
        ind = TRUE,
        ISB = FALSE,
        pdf = FALSE,
        col_den = c("#aa0078", "#0093af"))


# Plotting the measurements and estimates for 2000-2003 for RW model 2
mu.haringey4 <- jags.mod.fit.haringey4$BUGSoutput$mean$Z2
sd.haringey4 <- jags.mod.fit.haringey4$BUGSoutput$sd$Z2

# ```{r}
df.haringey8 <- data.frame(x4 = haringey3$Date, y7 = mu.haringey4, y8 = haringey3$Haringey,
                    lower.haringey4 = mu.haringey4 - 1.96*sd.haringey4,
                    upper.haringey4 = mu.haringey4 + 1.96*sd.haringey4)

ggplot(data = df.haringey8) +
  geom_point(aes(x = x4, y = y7), colour = 'grey31', size = 2) +
  geom_point(aes(x = x4, y = y8), colour = 'blue', size = 2) +
  geom_line(aes(x = x4, y = y7), colour = '#0093af', size = 1) +
  geom_line(aes(x = x4, y = lower.haringey4), linetype = "dashed", colour = "#aa0078", size = 1) +
  geom_line(aes(x = x4, y = upper.haringey4), linetype = "dashed", colour = "#aa0078", size = 1) +
  xlab('Date') + ylab('Z2 (estimates) and Haringey (measurements) of PM10 at Haringey') +
  ggtitle('PM10 measurements and estimates at Haringey for 2000-2003 with RW2 using informative priors from
Heathrow modelling') +
  theme(axis.title = element_text(size = 14),
        axis.text = element_text(size = 12),
        plot.title = element_text(size = 14))

# There is an improvement in which estimates converged for jags.mod.fit.haringey4 (using
# informative priors) compared to jags.mod.fit.haringey2 (using non-informative priors), which
# both employ RW2 models. Also, jags.mod.fit.haringey4 has a lower DIC value (= 8300.2) than
# for jags.mod.fit.haringey2 (= 13142.6). Output details from jags.mod.fit.haringey4 are given below:

# Trace plots were generated from jags.mcmc.haringey4 for each parameter (sigma2_19, sigma2_20,
# deviance, Z2[1] and selected examples of Z2 that covered both where measurements are known and
# where they are missing). The trace plots for deviance, sigma2_19 and sigma2_20 show some
# convergence for each chain, however there is some deviation from each other.
# For the Z2 estimates, the amount of convergence relies greatly on whether the parameter estimate
# has associated measurements in the haringey dataset or are missing values. The chains show minor
# convergence if the Z2 estimate has an associated haringey measurement, which is a vast improvement
# on jags.mod.fit.haringey2. However, the chains still do not converge if there are missing values.
# Further, the amount that a Z2 estimate does not converge becomes greater if the estimate
# is further from a Z2 estimate that has an associated measurement.

# In this way, Z2 estimates 975-1210 are missing data, but the convergence of Z2 estimate 1025 is
# worse than Z2 estimate 985, which is worse than Z2 estimate 975. Conversely, the convergence of
# Z2 estimate 1261 (with ass. measurement) is better than Z2 estimate 1219 (with ass. measurement),
# which is better than Z2 estimate 1211 (missing data), which again is better than Z2 estimate 1210
```

```
# (missing data). This shows that as each estimate is based on the one previous, there is a "memory"
# of previous Z2 estimates that reduces the convergence where estimates "go further into" blocks
# missing data and improves the convergence as the estimates become associated with measurements once more.



#### Informative priors from Heathrow data ####

## Reading in London pollution data
london_pollution <- read.csv("London_Pollution.csv")
london_pollution$Date <- as.Date(london_pollution$Date, format = "%d/%m/%Y")
head(london_pollution)
tail(london_pollution)
london_pollution[350:400,]
london_pollution[50:100,]

## The date and heathrow air pollution data are in the 2nd and 3rd columns, respectively,
## and there are 1827 measurements from 1st January 2000 to 31st December 2004:
haringey <- dplyr::select(london_pollution, Date, Haringey)
head(haringey)
tail(haringey)
count(haringey)
sapply(haringey, class)

# Creating new Day, Month and Year columns from Date column for haringey data
haringey <- haringey %>% separate(Date, into = c('Year', 'Month', 'Day'))
haringey2 <- cbind(haringey, Date = london_pollution$Date)
head(haringey2)
# arrange columns with Date, Day, Month, Year, haringey air pollution
haringey2 <- haringey2 %>% dplyr::select(Date, Year, Month, Day, Haringey)
head(haringey2)
# haringey data for just 2000-2003
haringey3 <- haringey2 %>% dplyr::filter(Year != 2004)
tail(haringey3)
haringey3[1001:1100,]
haringey3[1101:1200,]
haringey3[1201:1300,]



# RW1 model for Haringey data

# Set seed for reproducibility
set.seed(234)
# Set N to be the length of the data
N3 <- length(haringey3$Haringey)
N3
# Mean of haringey data
mean.haringey <- mean(haringey3$Haringey, na.rm = TRUE)
mean.haringey
# Extract haringey column data
Haringey <- haringey3$Haringey
Haringey

# From jags.mod.heathrow, sigma2_1 and sigma2_2 can inform what tau25 and tau26 are, respectively
sigma2_1 <- 27.534
tau25 <- 1/sigma2_1
```

```
sigma2_2 <- 25.013
tau26 <- 1/sigma2_2

# List the data to be used
jags.data.haringey5 <- list("Haringey", "N3")

# Model
jags.mod.haringey5 <- function(){
  Z2[1] ~ dnorm(0, 0.001)
  for(t in 2:N3){
    Haringey[t] ~ dnorm(Z2[t], tau25)            # normal likelihood of haringey data
    Z2[t] ~ dnorm(Z2[t-1], tau26)                # normal prediction model
  }
  # priors on measurement error and white noise variances
  tau25 ~ dnorm(0.036, 0.019)                    # normal measurement error precision model
  sigma2_25 <- 1/tau25                          # measurement error variance
  tau26 ~ dnorm(0.040, 0.040)                    # normal estimate error precision model
  sigma2_26 <- 1/tau26                          # estimate error variance
}

# Specify initial values
inits.haringey9 <- list("Z2[1]" = 22, "is.na(Haringey)" = mean.haringey, "tau25" = 0.036, "tau26" = 0.040)
inits.haringey10 <- list("Z2[1]" = 20, "is.na(Haringey)" = mean.haringey, "tau25" = 0.036, "tau26" = 0.040)
jags.inits.haringey5 <- list(inits.haringey9, inits.haringey10)

# Monitor the parameters to be used for the prediction
jags.param.haringey5 <- c("sigma2_25","sigma2_26","Z2")

# Fitting the new model
jags.mod.fit.haringey5 <- jags(data = jags.data.haringey5, inits = jags.inits.haringey5,
                      parameters.to.save = jags.param.haringey5, n.chains = 2,
                      n.iter = 10000, n.burnin = 5000, n.thin = 1,
                      model.file = jags.mod.haringey5)

# Get point and interval estimates
print(jags.mod.fit.haringey5)
# Inference for Bugs model at
"/var/folders/pc/hksslngn6_56y6dc8464zthm0000gn/T//RtmpSTiK0N/model24a3c22b237.txt", fit using jags,
# 2 chains, each with 10000 iterations (first 5000 discarded)
# n.sims = 10000 iteratipD = 14885.3 and DIC = 21437.0ons saved
#          mu.vect sd.vect    2.5%     25%     50%     75%    97.5%  Rhat n.eff
# Z2[1]     10.338   5.933  -1.220   6.388  10.386  14.374  21.811 1.001 10000
# Z2[2]     10.637   3.224   4.277   8.410  10.651  12.834  16.927 1.001 10000
# Z2[3]     10.046   2.904   4.561   8.056   9.974  12.034  15.750 1.001  4000
# Z2[4]     14.230   2.820   8.745  12.343  14.219  16.100  19.785 1.001  4300
# Z2[5]     18.528   2.864  12.984  16.571  18.536  20.495  24.165 1.003   700
# Z2[6]     20.154   2.865  14.572  18.209  20.169  22.091  25.763 1.002   990
# Z2[7]     21.962   2.864  16.399  19.975  21.978  23.891  27.567 1.002  2100
# Z2[8]     21.346   2.844  15.780  19.460  21.304  23.257  26.963 1.001  4900
# Z2[9]     24.298   2.834  18.659  22.428  24.332  26.215  29.886 1.001 10000
# Z2[10]    28.845   2.912  23.106  26.894  28.907  30.786  34.591 1.006   280
# For each parameter, n.eff is a crude measure of effective sample size,
# and Rhat is the potential scale reduction factor (at convergence, Rhat=1).

# DIC info (using the rule, pD = var(deviance)/2)
# pD = 14885.3 and DIC = 21437.0
# DIC is an estimate of expected predictive error (lower deviance is better).
```

```
# plot density manually
sim.values.haringey5 <- jags.mod.fit.haringey5$BUGSoutput$sims.list

df.haringey9 <- data.frame(sigma2_25 = sim.values.haringey5$sigma2_25,
                           sigma2_26 = sim.values.haringey5$sigma2_26)

haringey_sigma2_25 <- ggplot(data = df.haringey9, aes(x = sigma2_25)) +
  geom_density(colour = '#0093af', fill = '#0093af', alpha = 0.3) +
  theme(axis.title = element_blank(),
        axis.text = element_text(size = 12),
        plot.title = element_text(size = 14)) +
  geom_vline(xintercept = 0, linetype = "dashed",
             colour = "#aa0078", size = 1) +
  ggtitle('Posterior density of sigma2_25 - measurement variance')
haringey_sigma2_25

haringey_sigma2_26 <- ggplot(data = df.haringey9, aes(x = sigma2_26)) +
  geom_density(colour = '#0093af', fill = '#0093af', alpha = 0.3) +
  theme(axis.title = element_blank(),
        axis.text = element_text(size = 12),
        plot.title = element_text(size = 14)) +
  geom_vline(xintercept = 0, linetype = "dashed",
             colour = "#aa0078", size = 1) +
  ggtitle('Posterior density of sigma2_26 - estimate variance')
haringey_sigma2_26


# Create an MCMC object from the output of the haringey model
jags.mcmc.haringey5 <- as.mcmc(jags.mod.fit.haringey5)
# Graphical parameters
par(mar = c(2,4,4,2), cex = 1.0)
layout(matrix(c(1,1,2,2),2,2,byrow=TRUE))
# for a traceplot, producing for groups of 100 traceplots
traceplot(jags.mcmc.haringey5[,1401:1461], params = c("sigma2_25", "sigma2_26", "Z2"))
traceplot(jags.mcmc.haringey5[,1:50], params = c("sigma2_25", "sigma2_26", "Z2"))
traceplot(jags.mcmc.haringey5[,201:250], params = c("sigma2_25", "sigma2_26", "Z2"))


# for summary statistics
summary(jags.mcmc.haringey5[,1:5])
summary(jags.mcmc.haringey5[,201:250])
# Iterations = 5001:10000
# Thinning interval = 1
# Number of chains = 2
# Sample size per chain = 5000

# 1. Empirical mean and standard deviation for each variable, plus standard error of the mean:
#             Mean       SD  Naive SE Time-series SE
# deviance  6551.75 176.301  1.76301       13.37160
# sigma2_25   14.84   2.238  0.02238        0.16553
# sigma2_26   25.22   3.175  0.03175        0.22344
# Z2[1]       10.34   5.933  0.05933        0.08173
# Parameters with missing data
# Z2[1200] 27.27 16.245  0.16245        2.05506
# Z2[1201] 26.81 15.613  0.15613        1.72226
```

```
# Z2[1202] 26.35 14.930  0.14930       1.72814
# Z2[1203] 25.91 14.146  0.14146       1.35866
# Z2[1204] 25.43 13.240  0.13240       1.16545
# Z2[1205] 25.03 12.274  0.12274       1.09753
# Z2[1206] 24.70 11.222  0.11222       0.86707
# Z2[1207] 24.43 10.113  0.10113       0.60092
# Z2[1208] 24.04  8.964  0.08964       0.49194
# Z2[1209] 23.64  7.711  0.07711       0.34005
# Parameters with associated measurements
# Z2[1210] 23.24  5.961  0.05961       0.22468
# Z2[1211] 22.83  3.239  0.03239       0.07561
# Z2[1212] 17.07  2.900  0.02900       0.03947
# Z2[1213] 17.25  2.828  0.02828       0.03396
# Z2[1214] 13.88  2.837  0.02837       0.03665
# Z2[1215] 14.96  2.836  0.02836       0.03374
# Z2[1216] 15.02  2.807  0.02807       0.03298
# Z2[1217] 14.35  2.808  0.02808       0.03453
# Z2[1218] 13.36  2.826  0.02826       0.03335
# Z2[1219] 14.12  2.839  0.02839       0.03380

# 2. Quantiles for each variable:
#           2.5%    25%     50%     75%    97.5%
# deviance  6162.13 6440.996 6569.14 6676.70 6849.83
# sigma2_25   10.56   13.275   14.85   16.41   19.20
# sigma2_26   19.66   22.943   25.01   27.27   31.87
# Z2[1]      -1.22    6.388   10.39   14.37   21.81
# Parameters with associated measurements
# Z2[1200]  -3.1992 15.26 27.54 38.64 58.91
# Z2[1201]  -3.1458 15.49 27.14 37.90 56.90
# Z2[1202]  -2.5651 15.74 26.58 36.99 55.03
# Z2[1203]  -1.7998 16.22 25.97 36.06 53.00
# Z2[1204]  -0.1643 16.28 25.36 34.73 50.79
# Z2[1205]   1.3982 16.53 25.11 33.52 48.98
# Z2[1206]   2.8602 17.01 24.62 32.30 46.79
# Z2[1207]   5.0614 17.57 24.32 31.24 44.11
# Z2[1208]   6.7471 17.88 23.96 30.01 41.76
# Z2[1209]   8.9650 18.41 23.52 28.80 39.04
# Parameters with missing data
# Z2[1210]  11.8417 19.13 23.13 27.22 35.07
# Z2[1211]  16.4152 20.66 22.83 25.03 29.16
# Z2[1212]  11.5072 15.09 17.03 19.01 22.87
# Z2[1213]  11.7164 15.35 17.30 19.14 22.80
# Z2[1214]   8.3580 11.96 13.85 15.79 19.51
# Z2[1215]   9.4624 13.03 14.94 16.87 20.54
# Z2[1216]   9.5806 13.12 15.03 16.92 20.57
# Z2[1217]   8.9279 12.41 14.35 16.28 19.86
# Z2[1218]   7.6713 11.46 13.40 15.28 18.90
# Z2[1219]   8.5937 12.21 14.14 16.03 19.74


# Produce an MCMC trace of sigma2_25 output
MCMCtrace(jags.mcmc.haringey5,
        params = c('sigma2_25'),
        type = 'density',
        ind = TRUE,
        ISB = FALSE,
        pdf = FALSE,
```

```
        col_den = c("#aa0078", "#0093af"))

# Produce an MCMC trace of sigma2_26 output
MCMCtrace(jags.mcmc.haringey5,
        params = c('sigma2_26'),
        type = 'density',
        ind = TRUE,
        ISB = FALSE,
        pdf = FALSE,
        col_den = c("#aa0078", "#0093af"))

# Produce an MCMC trace of Z2 output
MCMCtrace(jags.mcmc.haringey5,
        params = c('Z2'),
        type = 'density',
        ind = TRUE,
        ISB = FALSE,
        pdf = FALSE,
        col_den = c("#aa0078", "#0093af"))

gelman.haringey5 <- gelman.diag(jags.mcmc.haringey5)
gelman.haringey5
# Potential scale reduction factors:
#           Point est. Upper C.I.
# deviance       1.10      1.35
# sigma2_25      1.07      1.28
# sigma2_26      1.09      1.34
# Z2[1]          1.00      1.00
# Parameters with missing data
# Z2[1200]       1.17      1.59
# Z2[1201]       1.14      1.50
# Z2[1202]       1.11      1.42
# Z2[1203]       1.09      1.36
# Z2[1204]       1.08      1.30
# Z2[1205]       1.05      1.23
# Z2[1206]       1.04      1.17
# Z2[1207]       1.03      1.15
# Z2[1208]       1.02      1.11
# Z2[1209]       1.02      1.07
# Parameters with associated measurements
# Z2[1210]       1.01      1.05
# Z2[1211]       1.00      1.02
# Z2[1212]       1.00      1.00
# Z2[1213]       1.00      1.00
# Z2[1214]       1.00      1.01
# Z2[1215]       1.00      1.00
# Z2[1216]       1.00      1.00
# Z2[1217]       1.00      1.00
# Z2[1218]       1.00      1.00
# Z2[1219]       1.00      1.00

# Multivariate psrf
# 4.24

# Create a data.frame from the gelman.diag output to put into a kable table
rhat.haringey5 <- as.data.frame((gelman.haringey5)$psrf)
kbl(rhat.haringey5) %>% kable_styling()
```

Below is the code to plot the Haringey measurements (NA values removed) against the Z2 estimates (including where
missing data has been estimated) and the associated upper and lower 95% credible intervals for the Z2 estimates.
Typically, the measurements and the Z2 estimates are similar to each other and the Heathrow measurements are within
the 95% credible intervals. Further, these credible intervals are tightly constrained the the Z2 estimates.
However, where there are missing data, the Z2 estimates deviate from the mean value of the Haringey measurements
and the credible interval expands greatly, so much so that the lower interval boundary is in negative numbers.
Of course, such numbers have no meaning as all measurements must be >0 (i.e. positive). Once there are measurements
to constrain the Z2 estimates, these estimates track closely to the measurements and the credible intervals move
to be close to the estimates once more. Consequently, this RW1 model fits the data well.

# ```{r}
mu.haringey5 <- jags.mod.fit.haringey5$BUGSoutput$mean$Z2
sd.haringey5 <- jags.mod.fit.haringey5$BUGSoutput$sd$Z2

# ```{r}
df.haringey10 <- data.frame(x5 = haringey3$Date, y9 = mu.haringey5, y10 = haringey3$Haringey,
                            lower.haringey5 = mu.haringey5 - 1.96*sd.haringey5,
                            upper.haringey5 = mu.haringey5 + 1.96*sd.haringey5)

ggplot(data = df.haringey10) +
  geom_point(aes(x = x5, y = y9), colour = 'grey31', size = 2) +
  geom_point(aes(x = x5, y = y10), colour = 'blue', size = 2) +
  geom_line(aes(x = x5, y = y9), colour = '#0093af', size = 1) +
  geom_line(aes(x = x5, y = lower.haringey5), linetype = "dashed", colour = "#aa0078", size = 1) +
  geom_line(aes(x = x5, y = upper.haringey5), linetype = "dashed", colour = "#aa0078", size = 1) +
  xlab('Date') + ylab('Z2 (estimates) and Haringey (measurements) of PM10 at Haringey') +
  ggtitle('PM10 measurements and estimates at Haringey for 2000-2003 for RW1 using informative priors from Heathrow modelling') +
  theme(axis.title = element_text(size = 14),
        axis.text = element_text(size = 12),
        plot.title = element_text(size = 14))

There is little or no improvement in which estimates converged for jags.mod.fit.haringey (using
non-informative priors) and jags.mod.fit.haringey5 (using informative priors), which both employ
RW1 models. However, jags.mod.fit.haringey3 has a lower DIC value (= 21437) than for
jags.mod.fit.haringey (= 41485.7). Details of the output from jags.mod.fit.haringey5 are given below:

Trace plots were generated from jags.mcmc.haringey5 for each parameter (sigma2_25, sigma2_26,
deviance, Z2[1] and selected examples of Z2 that covered both where measurements are known and
where they are missing). The trace plots for deviance, sigma2_25 and sigma2_26 are very close
to converging for each chain however there is a small amount deviation from each other.
For the Z2 estimates, the amount of convergence relies greatly on whether the parameter estimate
has associated measurements in the haringey dataset or are missing values. The chains converge
if the Z2 estimate has an associated haringey measurement and do not converge if there are missing
values. However, the amount that a Z2 estimate does not converge becomes greater if the estimate
is further from a Z2 estimate that has an associated measurement.

In this way, Z2 estimates 975-1210 are missing data, but the convergence of Z2 estimate 1025 is
worse than Z2 estimate 985, which is worse than Z2 estimate 975. Conversely, the convergence of
Z2 estimate 1261 (with ass. measurement) is better than Z2 estimate 1219 (with ass. measurement),
which is better than Z2 estimate 1211 (missing data), which again is better than Z2 estimate 1210

```
# (missing data). This shows that as each estimate is based on the one previous, there is a "memory"
# of previous Z2 estimates that reduces the convergence where estimates "go further into" blocks
# missing data and improves the convergence as the estimates become associated with measurements once more.




# RW2 model for Haringey data

# Set seed for reproducibility
set.seed(234)
# Set N to be the length of the data
N1 <- length(haringey3$Haringey)
N1
# Mean of haringey data
mean.haringey <- mean(haringey3$Haringey, na.rm = TRUE)
mean.haringey
# Extract haringey column data
Haringey <- haringey3$Haringey
Haringey

# From jags.mod.haringey2, sigma2_15 and sigma2_16 can inform what tau19 and tau20 are, repsectively
sigma2_3 <- 289.5
tau27 <- 1/sigma2_3
sigma2_4 <- 0.0476
tau28 <- 1/sigma2_4

# List the data to be used
jags.data.haringey6 <- list("Haringey", "N1")

# Model 2
jags.mod.haringey6 <- function(){
  Z2[1] ~ dnorm(0, 0.001)
  Z2[2] ~ dnorm(0, 0.001)
  for(t in 3:N1){
    Haringey[t] ~ dnorm(Z2[t], tau27)             # normal likelihood of haringey data
    Z2[t] ~ dnorm((2*Z2[t-1] - Z2[t-2]), tau28)     # normal prediction model
  }
  # informative priors on measurement error and white noise variances
  tau27 ~ dnorm(0.003, 0.0001)                  # normal measurement error precision model
  sigma2_27 <- 1/tau27                          # measurement error variance
  tau28 ~ dnorm(21, 100)                         # normal estimate error precision model
  sigma2_28 <- 1/tau28                          # estimate error variance
}

# Specify initial values
inits.haringey11 <- list("Z2[1]" = 22, "is.na(Haringey)" = mean.haringey, "tau27" = 0.003, "tau28" = 21)
inits.haringey12 <- list("Z2[1]" = 20, "is.na(Haringey)" = mean.haringey, "tau27" = 0.003, "tau28" = 21)
jags.inits.haringey6 <- list(inits.haringey11, inits.haringey12)

# Monitor the parameters to be used for the prediction
jags.param.haringey6 <- c("sigma2_27", "sigma2_28", "Z2")

# Fitting the new model
jags.mod.fit.haringey6 <- jags(data = jags.data.haringey6, inits = jags.inits.haringey6,
                               parameters.to.save = jags.param.haringey6, n.chains = 2,
                               n.iter = 10000, n.burnin = 5000, n.thin = 1,
```

```
                    model.file = jags.mod.haringey6)

# Get point and interval estimates
print(jags.mod.fit.haringey6)
# Inference for Bugs model at
"/var/folders/pc/hksslngn6_56y6dc8464zthm0000gn/T//Rtmpx3NKdV/model70b6e7da28b.txt", fit using jags,
# 2 chains, each with 10000 iterations (first 5000 discarded)
# n.sims = 10000 iterations saved
#          mu.vect sd.vect    2.5%     25%     50%     75%   97.5%  Rhat n.eff
# Z2[1]      5.447   8.761  -8.655  -3.175   5.804  12.910  19.179 4.535     2
# Z2[2]      5.305   7.998  -7.432  -2.659   5.574  12.221  17.335 4.788     2
# Z2[3]      5.163   7.263  -6.389  -2.061   5.929  11.521  15.576 5.035     2
# Z2[4]      5.021   6.564  -5.634  -1.402   6.071  10.849  13.822 5.233     2
# Z2[5]      4.875   5.906  -5.062  -0.741   6.080  10.256  12.248 5.312     2
# Z2[6]      4.712   5.298  -4.547  -0.157   5.712   9.620  11.307 5.220     2
# Z2[7]      4.529   4.747  -4.148   0.301   5.147   8.793  10.722 4.972     2
# Z2[8]      4.326   4.254  -3.732   0.745   5.039   8.027  10.159 4.621     2
# Z2[9]      4.107   3.812  -3.249   1.082   4.758   7.301   9.639 4.245     2
# Z2[10]     3.878   3.411  -2.872   1.257   4.317   6.628   9.135 3.895     2

# For each parameter, n.eff is a crude measure of effective sample size,
# and Rhat is the potential scale reduction factor (at convergence, Rhat=1).

# DIC info (using the rule, pD = var(deviance)/2)
# pD = 7933.7 and DIC = 18114.9
# DIC is an estimate of expected predictive error (lower deviance is better).


# Create an MCMC object from the output of the haringey model
jags.mcmc.haringey6 <- as.mcmc(jags.mod.fit.haringey6)
# Graphical parameters
par(mar = c(2,4,4,2), cex = 1.0)
layout(matrix(c(1,1,2,2),2,2,byrow=TRUE))
# for a traceplot, producing for groups of 100 traceplots
traceplot(jags.mcmc.haringey6[,1401:1461], params = c("sigma2_27", "sigma2_28", "Z2"))
traceplot(jags.mcmc.haringey6[,1:10], params = c("sigma2_27", "sigma2_28", "Z2"))
traceplot(jags.mcmc.haringey6[,201:250], params = c("sigma2_27", "sigma2_28", "Z2"))

# for summary statistics
summary(jags.mcmc.haringey6[,1:5])
summary(jags.mcmc.haringey6[,201:250])
# Iterations = 5001:10000
# Thinning interval = 1
# Number of chains = 2
# Sample size per chain = 5000

# 1. Empirical mean and standard deviation for each variable, plus standard error of the mean:
#              Mean      SD  Naive SE  Time-series SE
# deviance  1.018e+04 1.302e+02 1.302e+00     6.821e+01
# sigma2_27 3.170e+02 3.686e+01 3.686e-01     1.482e+01
# sigma2_28 4.759e-02 2.249e-04 2.249e-06     2.857e-06
# Z2[1]     5.447e+00 8.761e+00 8.761e-02     1.441e+00
# Parameters with missing data
# Z2[1200] -0.2567 1.776  0.01776       0.6993
# Z2[1201]  0.0349 1.740  0.01740       0.8535
# Z2[1202]  0.3326 1.701  0.01701       0.7968
# Z2[1203]  0.6273 1.649  0.01649       0.6613
```

```
# Z2[1204]  0.9100 1.576  0.01576       0.5903
# Z2[1205]  1.1832 1.484  0.01484       0.4874
# Z2[1206]  1.4464 1.388  0.01388       0.4082
# Z2[1207]  1.6967 1.309  0.01309       0.3069
# Z2[1208]  1.9255 1.263  0.01263       0.2692
# Z2[1209]  2.1245 1.254  0.01254       0.2471
# Parameters with associated measurements
# Z2[1210]  2.2961 1.278  0.01278       0.2650
# Z2[1211]  2.4405 1.319  0.01319       0.3060
# Z2[1212]  2.5526 1.356  0.01356       0.3459
# Z2[1213]  2.6365 1.372  0.01372       0.3899
# Z2[1214]  2.6871 1.365  0.01365       0.3307
# Z2[1215]  2.7019 1.343  0.01343       0.3367
# Z2[1216]  2.6890 1.318  0.01318       0.2979
# Z2[1217]  2.6491 1.300  0.01300       0.2698
# Z2[1218]  2.5891 1.308  0.01308       0.2240
# Z2[1219]  2.5175 1.353  0.01353       0.2067

# 2. Quantiles for each variable:
#              2.5%      25%      50%      75%     97.5%
# deviance  9924.91807  1.007e+04 1.020e+04 1.029e+04 1.039e+04
# sigma2_27  250.59756  2.876e+02 3.186e+02 3.451e+02 3.854e+02
# sigma2_28    0.04715  4.743e-02 4.759e-02 4.775e-02 4.802e-02
# Z2[1]      -8.65472 -3.175e+00 5.804e+00 1.291e+01 1.918e+01
# Parameters with missing data
# Z2[1200] -3.85757 -1.3097 -0.2442  0.92146 2.901
# Z2[1201] -3.74579 -0.9853  0.1106  1.23015 3.045
# Z2[1202] -3.48848 -0.6639  0.4814  1.49663 3.205
# Z2[1203] -3.10944 -0.3908  0.7972  1.80161 3.410
# Z2[1204] -2.58594 -0.1275  1.0704  2.06794 3.563
# Z2[1205] -1.96785  0.0994  1.3238  2.27971 3.666
# Z2[1206] -1.28767  0.4500  1.5632  2.45216 3.778
# Z2[1207] -0.69661  0.6523  1.7469  2.71073 3.976
# Z2[1208] -0.39929  0.9604  1.9281  2.93512 4.100
# Z2[1209] -0.21727  1.1612  2.1714  3.12721 4.312
# Parameters with associated measurements
# Z2[1210] -0.08502  1.2973  2.3425  3.30082 4.647
# Z2[1211]  0.07825  1.4220  2.4542  3.52310 4.942
# Z2[1212]  0.16520  1.4697  2.5150  3.63678 5.059
# Z2[1213]  0.21238  1.5348  2.5774  3.77766 5.132
# Z2[1214]  0.26555  1.5854  2.5912  3.83608 5.283
# Z2[1215]  0.29231  1.6330  2.6128  3.82141 5.357
# Z2[1216]  0.29164  1.6495  2.6450  3.74931 5.270
# Z2[1217]  0.26152  1.6422  2.6747  3.63342 5.151
# Z2[1218]  0.12647  1.5877  2.6576  3.52139 5.086
# Z2[1219] -0.02832  1.4373  2.7005  3.46358 5.053

gelman.haringey6 <- gelman.diag(jags.mcmc.haringey6)
gelman.haringey6
# Potential scale reduction factors:
#           Point est. Upper C.I.
# deviance        1.13       1.47
# sigma2_27       1.13       1.46
# sigma2_28       1.00       1.00
# Z2[1]           4.53      10.34
# Parameters with missing data
# Z2[1200]        1.20       1.71
```

```
# Z2[1201]      1.13     1.31
# Z2[1202]      1.10     1.10
# Z2[1203]      1.09     1.17
# Z2[1204]      1.13     1.44
# Z2[1205]      1.23     1.83
# Z2[1206]      1.40     2.32
# Z2[1207]      1.59     2.81
# Z2[1208]      1.73     3.14
# Z2[1209]      1.77     3.20
# Parameters with associated measurements
# Z2[1210]      1.71     3.03
# Z2[1211]      1.61     2.76
# Z2[1212]      1.52     2.53
# Z2[1213]      1.47     2.40
# Z2[1214]      1.46     2.37
# Z2[1215]      1.50     2.48
# Z2[1216]      1.61     2.76
# Z2[1217]      1.80     3.29
# Z2[1218]      2.11     4.11
# Z2[1219]      2.54     5.18

# Multivariate psrf
# 175

# Create a data.frame from the gelman.diag output to put into a kable table
rhat.haringey6 <- as.data.frame((gelman.haringey6)$psrf)
kbl(rhat.haringey6) %>% kable_styling()



# plot density manually
sim.values.haringey6 <- jags.mod.fit.haringey6$BUGSoutput$sims.list

df.haringey11 <- data.frame(sigma2_27 = sim.values.haringey6$sigma2_27,
                            sigma2_28 = sim.values.haringey6$sigma2_28)

haringey_sigma2_27 <- ggplot(data = df.haringey11, aes(x = sigma2_27)) +
  geom_density(colour = '#0093af', fill = '#0093af', alpha = 0.3) +
  theme(axis.title = element_blank(),
        axis.text = element_text(size = 12),
        plot.title = element_text(size = 14)) +
  geom_vline(xintercept = 0, linetype = "dashed",
             colour = "#aa0078", size = 1) +
  ggtitle('Posterior density of sigma2_27 - measurement variance')
haringey_sigma2_27

haringey_sigma2_28 <- ggplot(data = df.haringey11, aes(x = sigma2_28)) +
  geom_density(colour = '#0093af', fill = '#0093af', alpha = 0.3) +
  theme(axis.title = element_blank(),
        axis.text = element_text(size = 12),
        plot.title = element_text(size = 14)) +
  geom_vline(xintercept = 0, linetype = "dashed",
             colour = "#aa0078", size = 1) +
  ggtitle('Posterior density of sigma2_28 - estimate variance')
haringey_sigma2_28
```

```
# Produce an MCMC trace of sigma2_27 output
MCMCtrace(jags.mcmc.haringey6,
        params = c('sigma2_27'),
        type = 'density',
        ind = TRUE,
        ISB = FALSE,
        pdf = FALSE,
        col_den = c("#aa0078", "#0093af"))

# Produce an MCMC trace of sigma2_28 output
MCMCtrace(jags.mcmc.haringey6,
        params = c('sigma2_28'),
        type = 'density',
        ind = TRUE,
        ISB = FALSE,
        pdf = FALSE,
        col_den = c("#aa0078", "#0093af"))

# Produce an MCMC trace of Z2 output
MCMCtrace(jags.mcmc.haringey6,
        params = c('Z2'),
        type = 'density',
        ind = TRUE,
        ISB = FALSE,
        pdf = FALSE,
        col_den = c("#aa0078", "#0093af"))


# Plotting the measurements and estimates for 2000-2003 for RW model 2
mu.haringey6 <- jags.mod.fit.haringey6$BUGSoutput$mean$Z2
sd.haringey6 <- jags.mod.fit.haringey6$BUGSoutput$sd$Z2

# ```{r}
df.haringey12 <- data.frame(x6 = haringey3$Date, y11 = mu.haringey6, y12 = haringey3$Haringey,
                    lower.haringey6 = mu.haringey6 - 1.96*sd.haringey6,
                    upper.haringey6 = mu.haringey6 + 1.96*sd.haringey6)

ggplot(data = df.haringey12) +
  geom_point(aes(x = x6, y = y11), colour = 'grey31', size = 2) +
  geom_point(aes(x = x6, y = y12), colour = 'blue', size = 2) +
  geom_line(aes(x = x6, y = y11), colour = '#0093af', size = 1) +
  geom_line(aes(x = x6, y = lower.haringey6), linetype = "dashed", colour = "#aa0078", size = 1) +
  geom_line(aes(x = x6, y = upper.haringey6), linetype = "dashed", colour = "#aa0078", size = 1) +
  xlab('Date') + ylab('Z2 (estimates) and Haringey (measurements) of PM10 at Haringey') +
  ggtitle('PM10 measurements and estimates at Haringey for 2000-2003 with RW2 using informative priors from
Heathrow modelling') +
  theme(axis.title = element_text(size = 14),
        axis.text = element_text(size = 12),
        plot.title = element_text(size = 14))

# There is no real improvement in which estimates converged for jags.mod.fit.haringey6 (using
# informative priors) compared to jags.mod.fit.haringey2 (using non-informative priors), which
# both employ RW2 models. Also, jags.mod.fit.haringey6 has a higher DIC value (= 18114.9) than
# for jags.mod.fit.haringey2 (= 13142.6). Output details from jags.mod.fit.haringey4 are given below:

# Trace plots were generated from jags.mcmc.haringey6 for each parameter (sigma2_27, sigma2_28,
# deviance, Z2[1] and selected examples of Z2 that covered both where measurements are known and
```

where they are missing). The trace plots for deviance and sigma2_27 show convergence but for
sigma2_28 there is some convergence for each chain. For the Z2 estimates, there is no convergence
whether there are recorded measurements or missing data in the haringey dataset.