Report 2: Variance models for protein expression with missing data.

Salvador Garcia, s1655274

April 14th 2017

1 Description of the problem

Concentration of 460 proteins was measured for each of 9 rats using images of 2D gels. Sensitivity of the equipment resulted in concentration values less than 5 to be treated as missing. The aim is to create an appropriate model for modelling variability of the concentration of proteins and of the missing data.

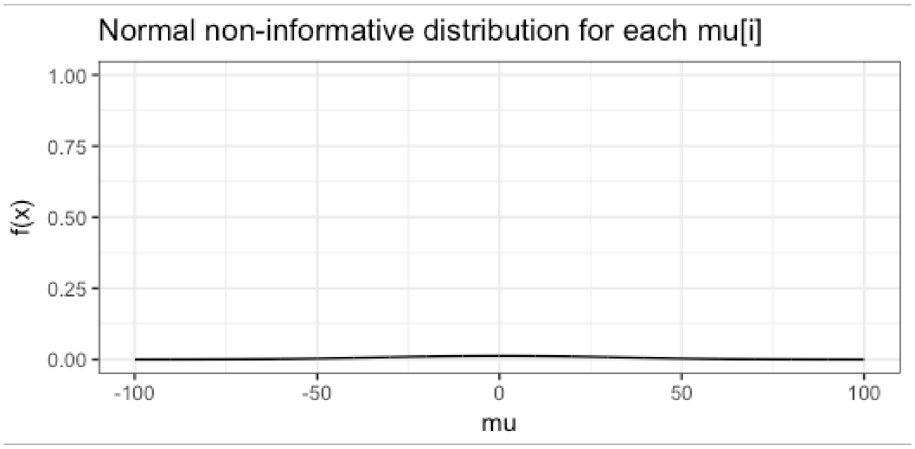
Let X_{gr} be the observed concentration of protein g in rat r, $g = 1; ...; 460$ and $r = 1; ...; 9$. It is known that the (unobserved) true concentration is different for different proteins however not much is known about their variability. A reasonable assumption is that their variability is similar or the same. Typically, logarithm of the concentration X_{gr} is assumed to have a normal distribution: $Y_{gr} = \log X_{gr}$,

$$Y_{gr}|\mu_g\sigma_g^2 \sim N(\mu_g, \sigma_g^2)$$

2 Modelling the variance

Non-informative prior for μ_g

For the prior of μ_g I am going to use a normal prior with high variance $N(0, 1000)$ (As suggested in the lectures). Remembering, if we want to model the effect independently in each protein, we would like to use a vague prior.



Bayesian models for the variance σ_g^2

Three different models will be used for the variance. The first assumes equal variance for all the proteins $\sigma_g^2 = \sigma^2$ with a non-informative prior for σ^2 (In the code the prior was for τ). The second and third are a hierarchical model for protein specific variance considering a log-normal and a gamma family each one.

· · · · · Equal variance for all proteins $\sigma_g^2 = \sigma^2$ (model 1) · · · · ·

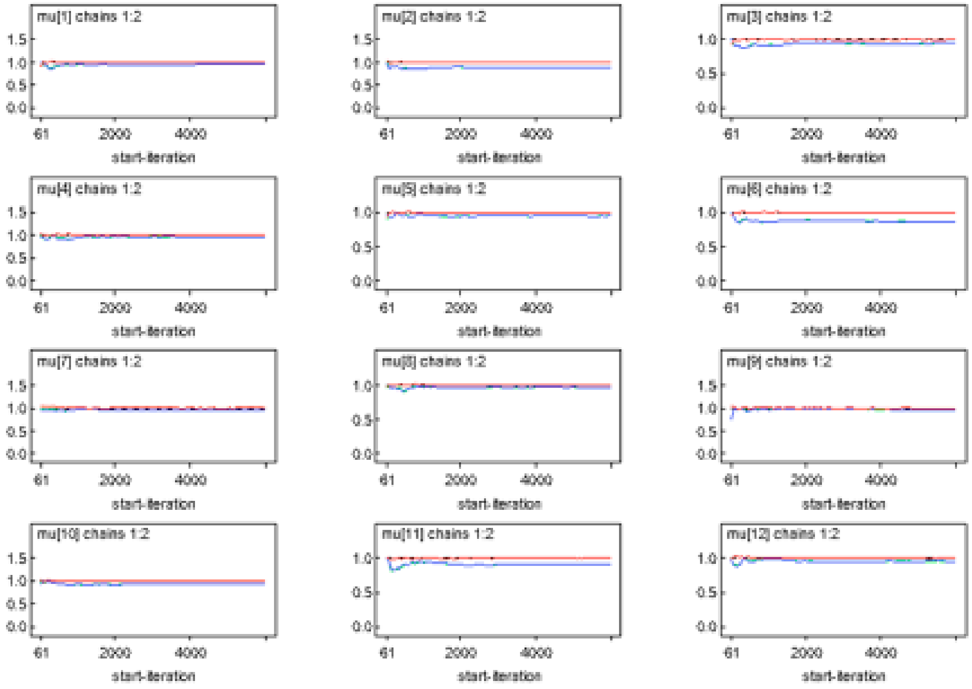
Burning, thinning and number of iterations For the first model 12000 iterations were done with a burning of 1001. No thinning was necessary. The inits for the first chain in the model were as follows: each μ_i was set with a value of 0, and $\tau = 1$. In the second chain each μ_i had the value of 7 of $\tau = 6$.

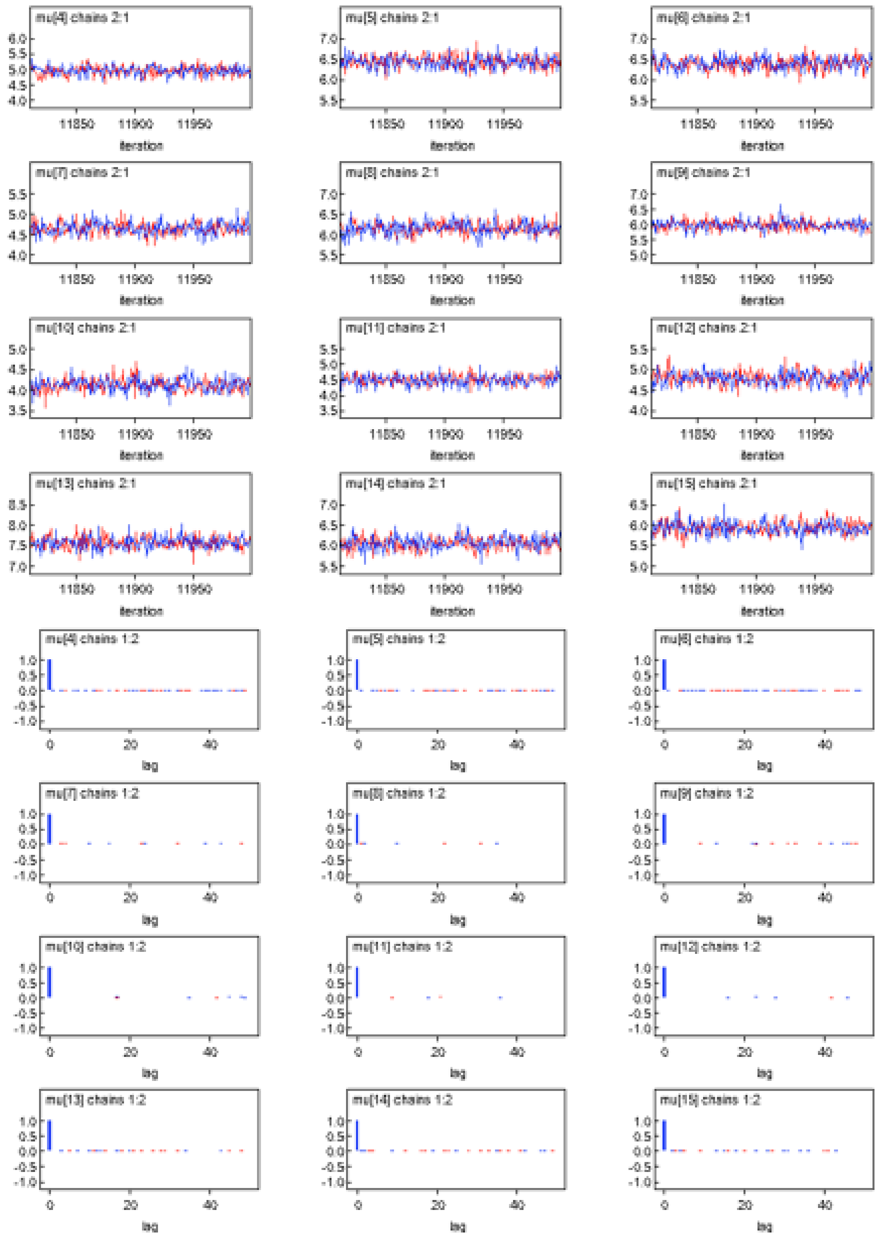
Table 1: Criterias for the model

Criteria	Value
Number of iterations	12000
Thinning	1
Burning	1001
Number of chains	2

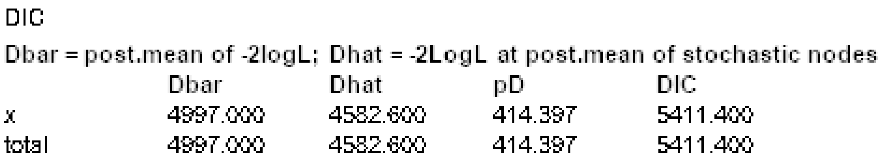
Model convergence and DIC

First, I will analyze the convergence of the parameters μ_i. To do that, the traceplots of the last 200 iterations (although all history was analyzed), the plots of the autocorrelations and the plots of the bgr diagnosis are show for the first 16 parameters μ_i (there are 413):



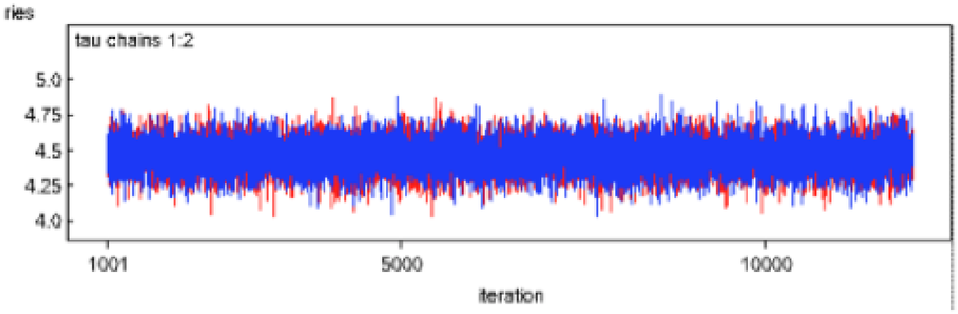


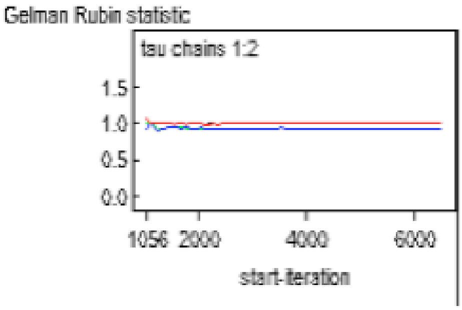
These plots show that there are no problems with autocorrelation or convergence. The DIC for this model is the following:

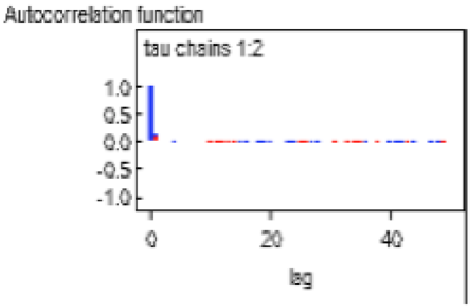


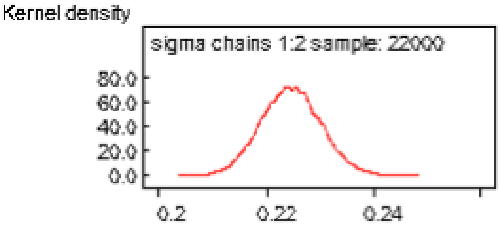
```
DIC
Dbar = post.mean of -2logL; Dhat = -2LogL at post.mean of stochastic nodes
          Dbar        Dhat        pD        DIC
x         4997.000    4582.600    414.397   5411.400
total     4997.000    4582.600    414.397   5411.400
```

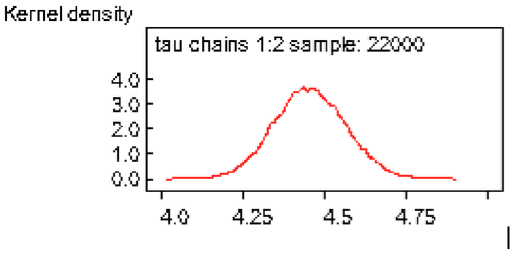
The important part of the first section in the report is to model the variance. In this first model just one τ is assumed, and the prior for this parameter is a non-informative $Gamma(.01, .01)$ distribution. Now are presented the plots of the σ and τ parameters. We can see that there is no problem of convergence or autocorrelation for these parameters.

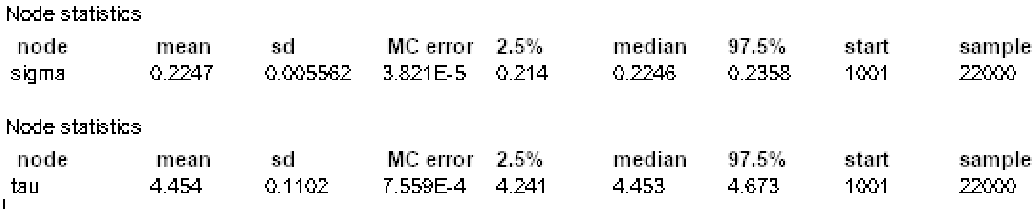












```
Node statistics
 node      mean     sd        MC error   2.5%     median    97.5%    start    sample
 sigma     0.2247   0.005562  3.821E-5   0.214    0.2246    0.2358   1001     22000

Node statistics
 node      mean     sd        MC error   2.5%     median    97.5%    start    sample
 tau       4.454    0.1102    7.559E-4   4.241    4.453     4.673    1001     22000
```

· · · · · Hierarchical models for protein specific variance $\sigma_g^2|\theta \sim p(\theta)$ · · · · ·

For this subsection two families of distributions will be used to model. In the first a gamma distribution will be used as a prior for τ and in the second a lognormal distribution will be assumed for σ.

Using a gamma distribution for the hyperparameters (model2)

Burning, thinning and number of iterations

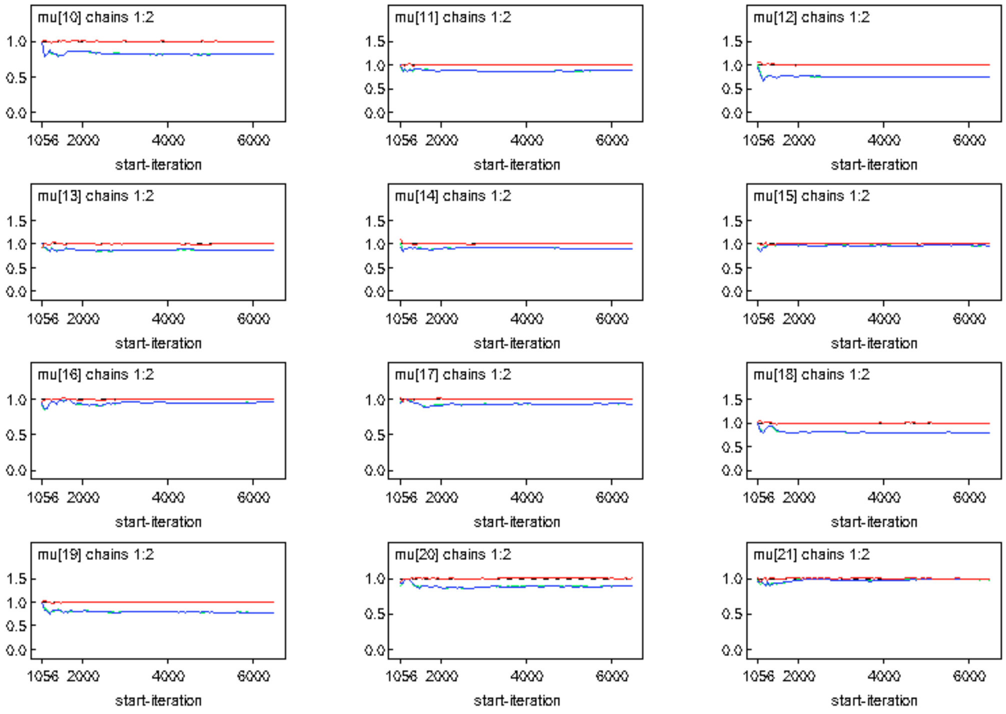
For the second model, 12000 iterations were done with a burning of 1001. Again, no thinning was necessary. The inits for the first chain in the model were as follows: each μ_i was set with a value of 0, and each $\tau_i = 1$. In the second chain each μ_i had the value of 6 of $\tau = 7$.

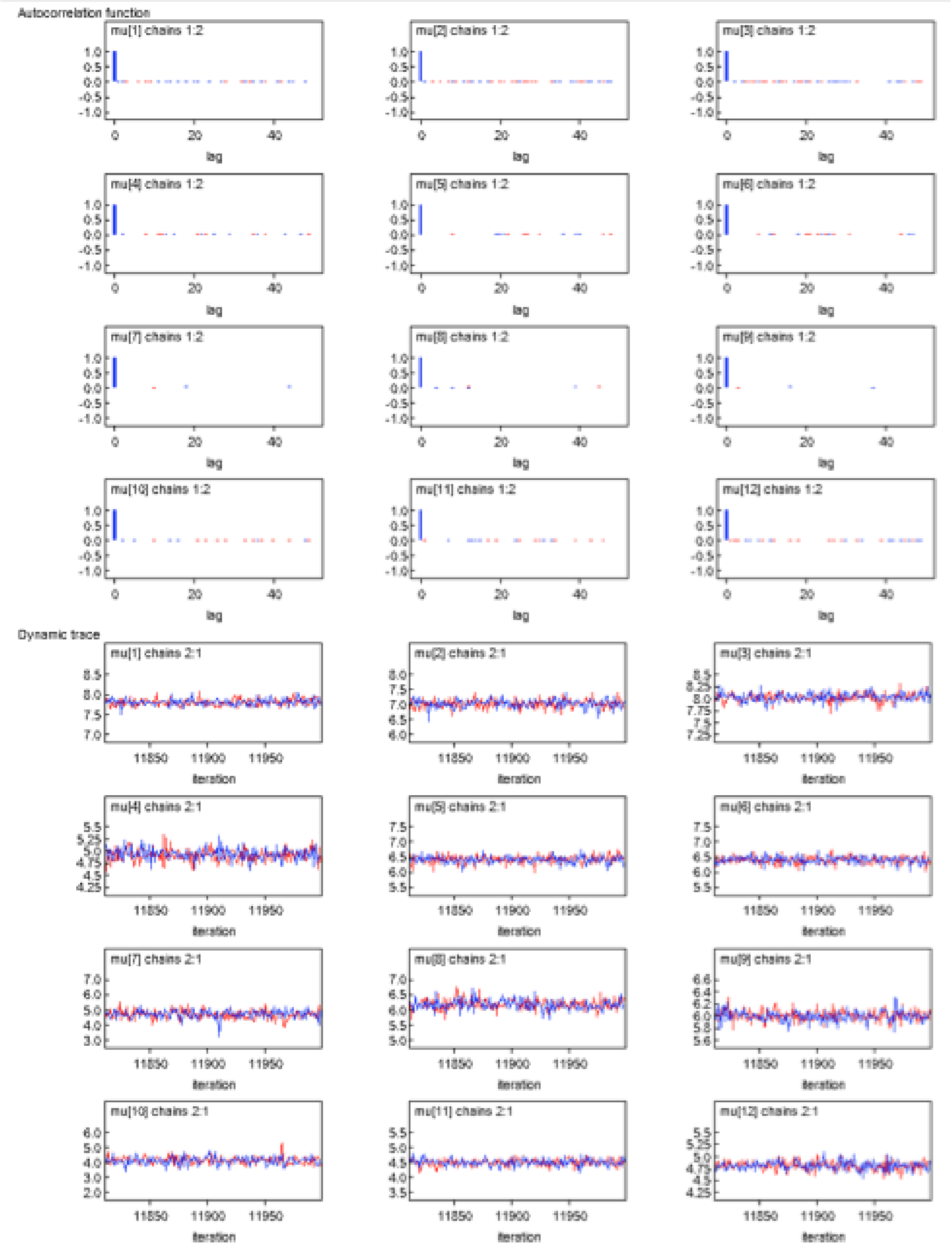
Table 2: Criterias for the model

Criteria	Value
Number of iterations	12000
Thinning	1
Burning	1001
Number of chains	2

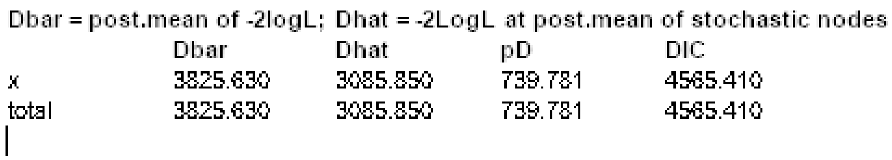
Model convergence and DIC

Now, I will analyze the convergence of the parameters μ_i. To do that, the traceplots of the last 200 iterations (although all history was analyzed), the plots of the autocorrelations and the plots of the bgr diagnosis are show for the first 16 parameters μ_i (there are 413):



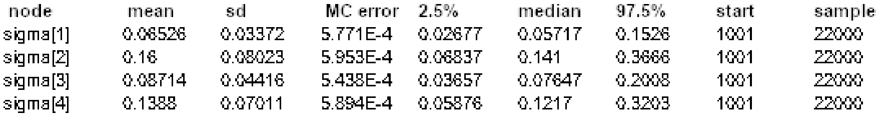


These plots show that there are no problems with autocorrelation or convergence. The DIC for this model is the following:

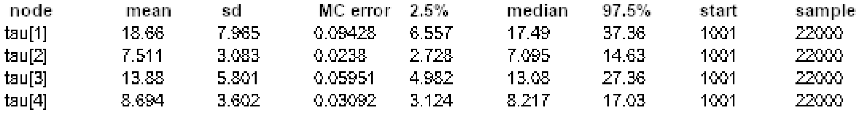


```
Dbar = post.mean of -2logL; Dhat = -2LogL at post.mean of stochastic nodes
                Dbar        Dhat        pD          DIC
x               3825.630    3085.850    739.781     4565.410
total           3825.630    3085.850    739.781     4565.410
```

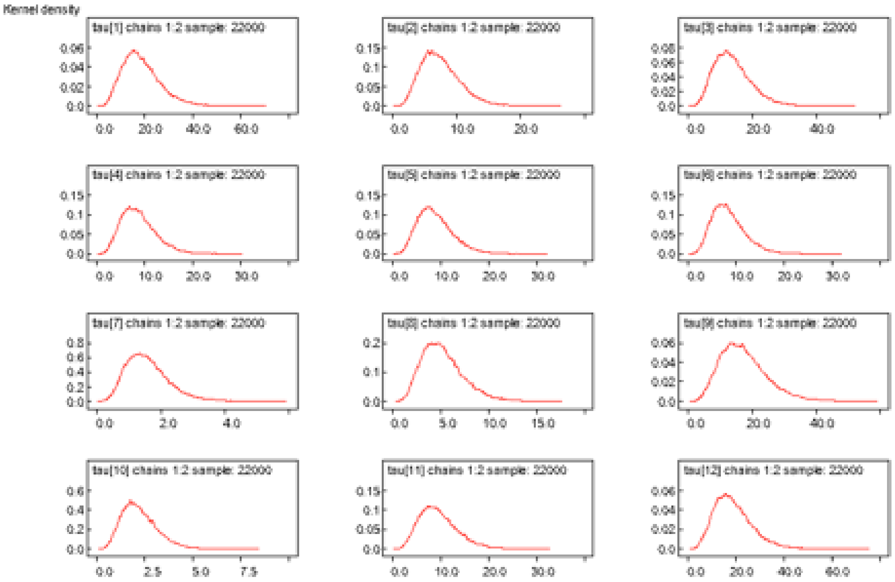
In this second model each row has a different τ_i parameter, and the prior for this parameters are a $Gamma(a, b)$ distribution, with a, b hyperparameters with gamma non informative distribution. Now are presented the plots of the σ and τ parameters. We can see that there is no problem of convergence or autocorrelation for these parameters:

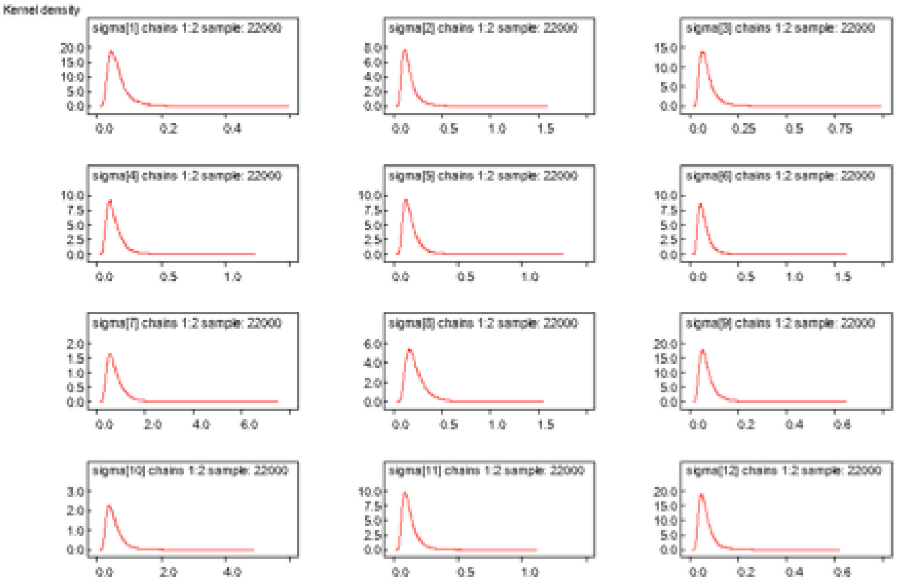


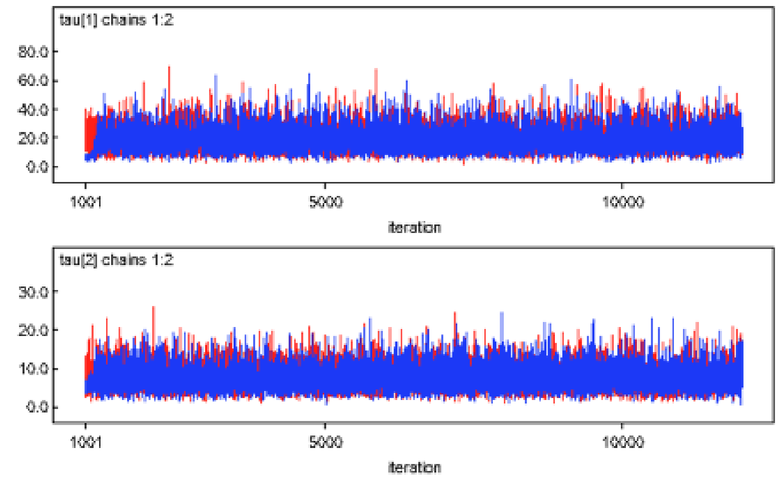
node	mean	sd	MC error	2.5%	median	97.5%	start	sample
sigma[1]	0.06526	0.03372	5.771E-4	0.02677	0.05717	0.1526	1001	22000
sigma[2]	0.16	0.08023	5.953E-4	0.06837	0.141	0.3666	1001	22000
sigma[3]	0.08714	0.04416	5.438E-4	0.03657	0.07647	0.2008	1001	22000
sigma[4]	0.1388	0.07011	5.894E-4	0.05876	0.1217	0.3203	1001	22000

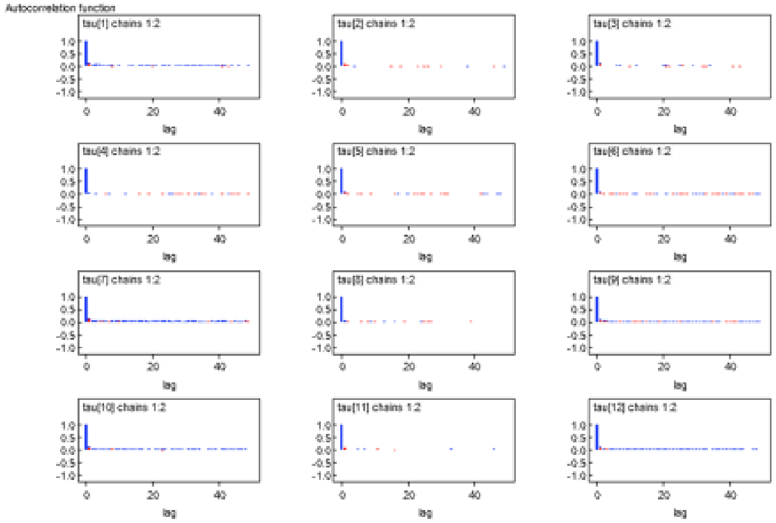


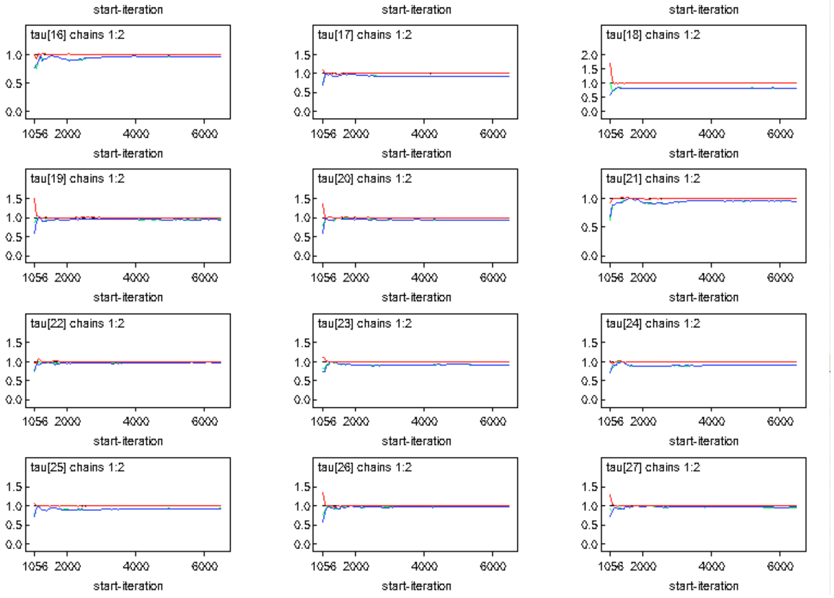
node	mean	sd	MC error	2.5%	median	97.5%	start	sample
tau[1]	18.66	7.965	0.09428	6.557	17.49	37.36	1001	22000
tau[2]	7.511	3.083	0.0238	2.728	7.095	14.63	1001	22000
tau[3]	13.88	5.801	0.05951	4.982	13.08	27.36	1001	22000
tau[4]	8.694	3.602	0.03092	3.124	8.217	17.03	1001	22000






tau[1] chains 1:2
80.0
60.0
40.0
20.0
0.0
1001
5000
10000
iteration
tau[2] chains 1:2
30.0
20.0
10.0
0.0
1001
5000
10000
iteration


Autocorrelation function
tau[1] chains 1:2
tau[2] chains 1:2
tau[3] chains 1:2
tau[4] chains 1:2
tau[5] chains 1:2
tau[6] chains 1:2
tau[7] chains 1:2
tau[8] chains 1:2
tau[9] chains 1:2
tau[10] chains 1:2
tau[11] chains 1:2
tau[12] chains 1:2
1.0
0.5
0.0
-0.5
-1.0
0
20
40
lag


start-iteration
tau[16] chains 1:2
tau[17] chains 1:2
tau[18] chains 1:2
tau[19] chains 1:2
tau[20] chains 1:2
tau[21] chains 1:2
tau[22] chains 1:2
tau[23] chains 1:2
tau[24] chains 1:2
tau[25] chains 1:2
tau[26] chains 1:2
tau[27] chains 1:2
2.0
1.5
1.0
0.5
0.0
1056
2000
4000
6000
start-iteration

Using a lognormal distribution for the hyperparameters (model3) For the third model, 15000 iterations were done with a burning of 2001. Again, no thinning was necessary. The inits for the first chain in the model were as follows: each μ_i was set with a value of 0, and each $\tau_i = 1$. In the second chain each μ_i had the value of 6 of $\tau = 7$.

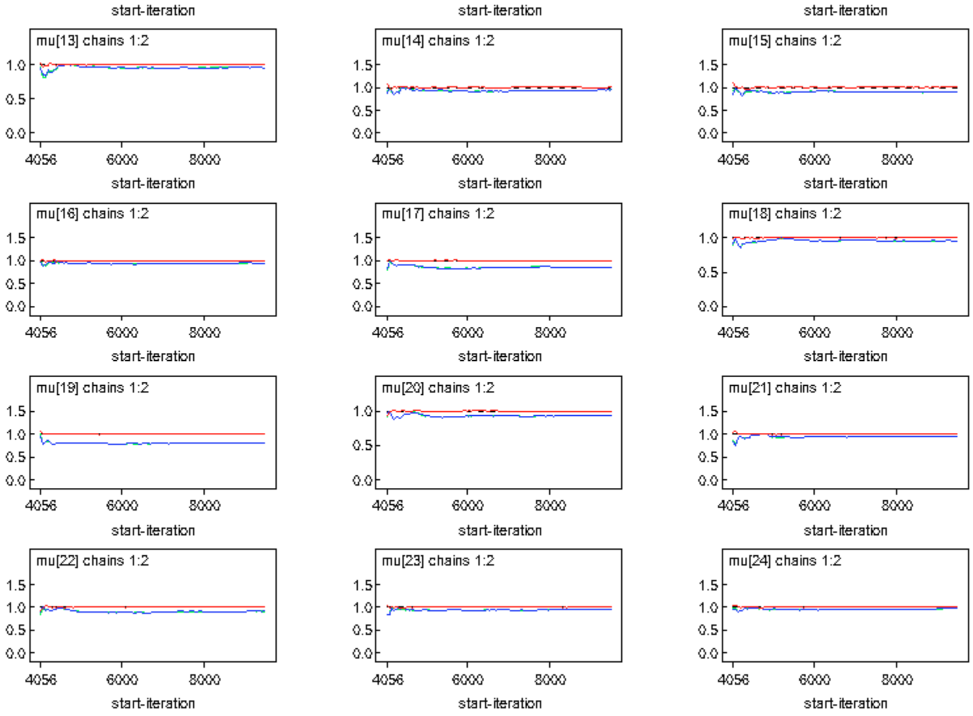
Burning, thinning and number of iterations

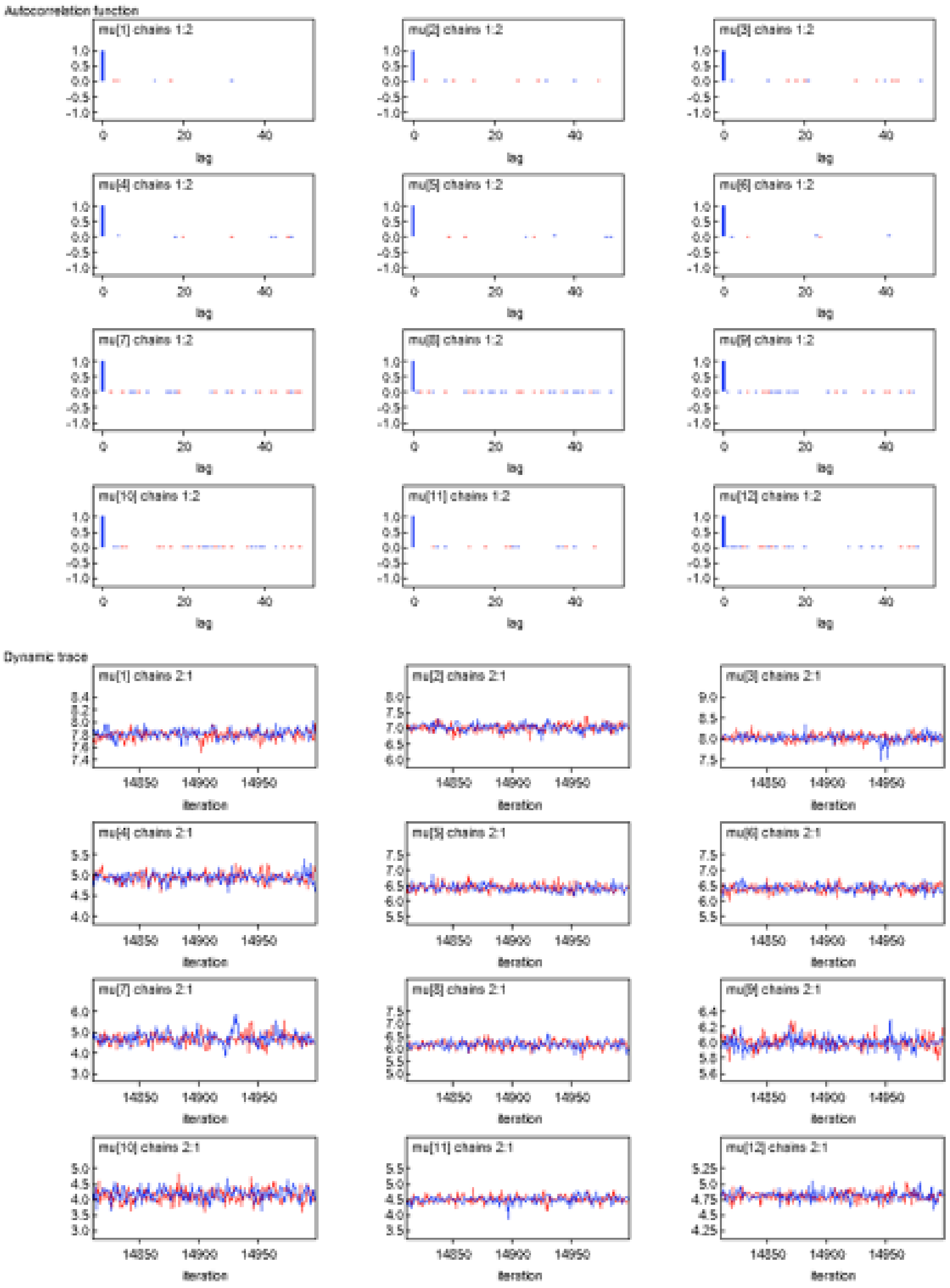
Table 3: Criterias for the model

Criteria	Value
Number of iterations	6000
Thinning	10
Burning	501
Number of chains	2

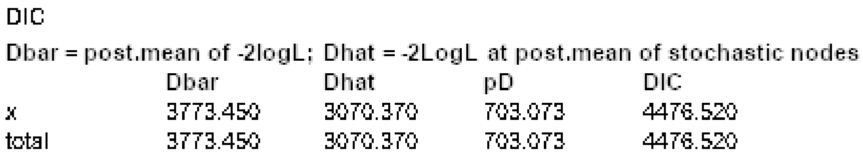
Model convergence and DIC

Now, I will analyze the convergence of the parameters μ_i. To do that, the traceplots of the last 200 iterations (although all history was analyzed), the plots of the autocorrelations and the plots of the bgr diagnosis are show for the first 16 parameters μ_i (there are 413):





These plots show that there are no problems with autocorrelation or convergence. The DIC for this model is the following:

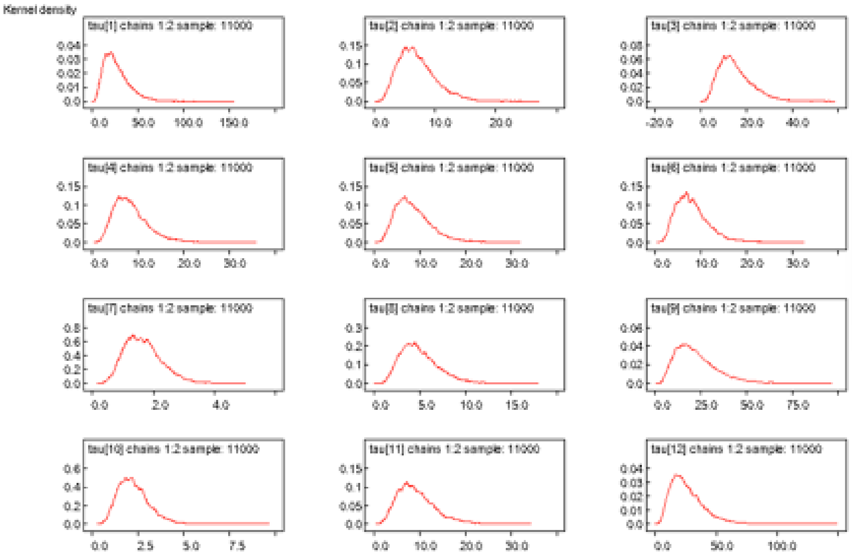


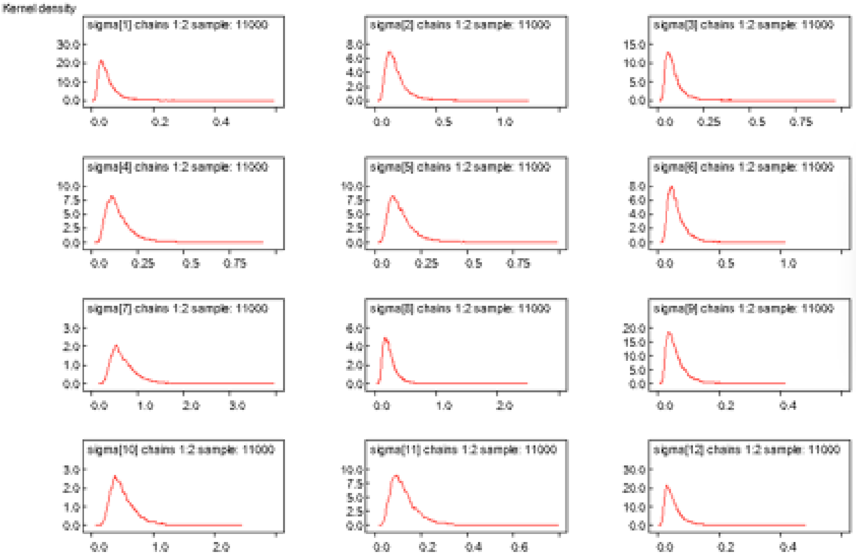
```
DIC
Dbar = post.mean of -2logL; Dhat = -2LogL at post.mean of stochastic nodes
           Dbar        Dhat        pD         DIC
x          3773.450    3070.370    703.073    4476.520
total      3773.450    3070.370    703.073    4476.520
```

Now, the parameters τ_i and σ_i will be analized (just some of them):

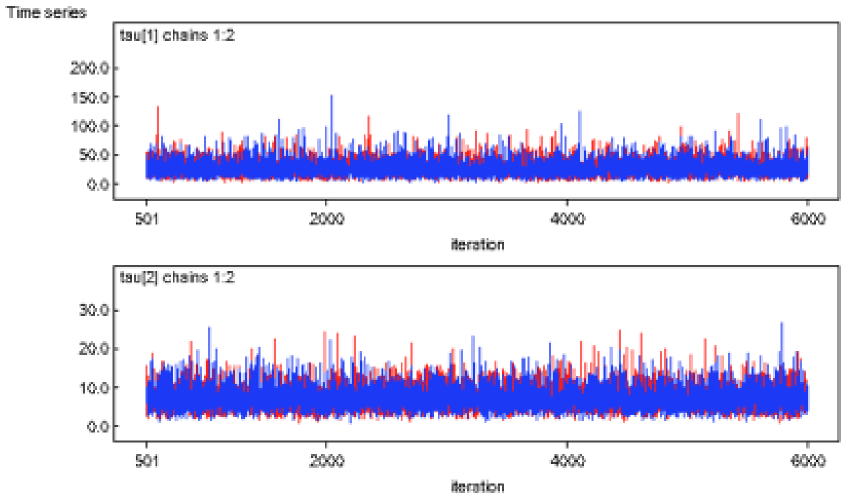
node	mean	sd	MC error	2.5%	median	97.5%	start	sample
sigma[1]	0.05135	0.03316	3.451E-4	0.01644	0.04284	0.14	501	11000
sigma[2]	0.1719	0.08887	8.553E-4	0.06956	0.1505	0.4015	501	11000
sigma[3]	0.08495	0.05056	4.762E-4	0.03052	0.07272	0.2114	501	11000
sigma[4]	0.1469	0.0752	7.753E-4	0.05803	0.129	0.3424	501	11000

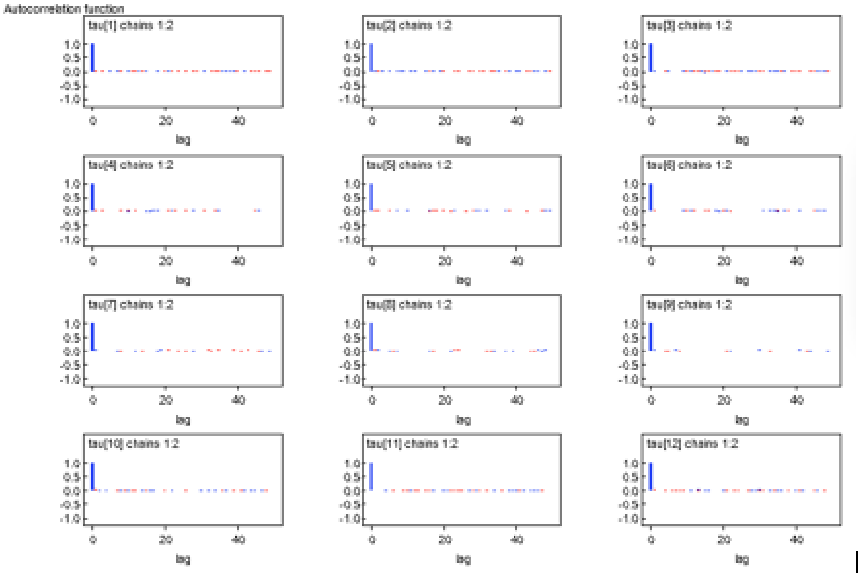
node	mean	sd	MC error	2.5%	median	97.5%	start	sample
tau[1]	26.23	14.15	0.1335	7.153	23.34	60.87	501	11000
tau[2]	7.129	3.123	0.02889	2.491	6.645	14.39	501	11000
tau[3]	15.14	7.421	0.07255	4.73	13.76	32.78	501	11000
tau[4]	8.337	3.672	0.03668	2.924	7.75	17.23	501	11000

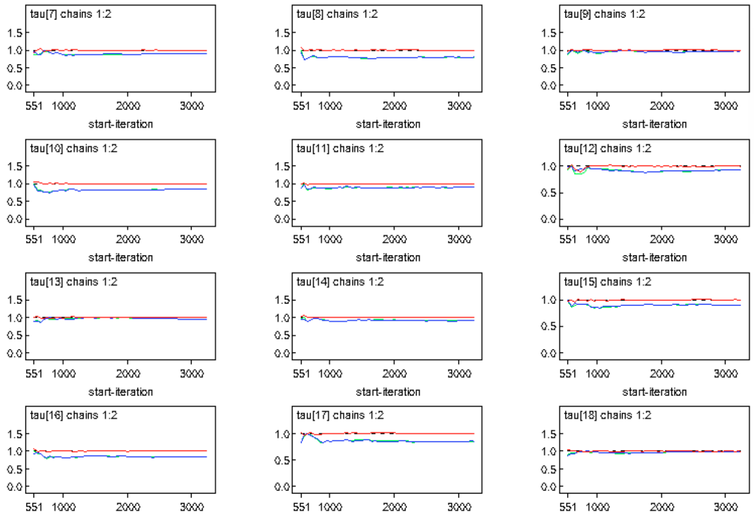




The following plots show no convergence problem. Also these does not show autocorrelation problems.

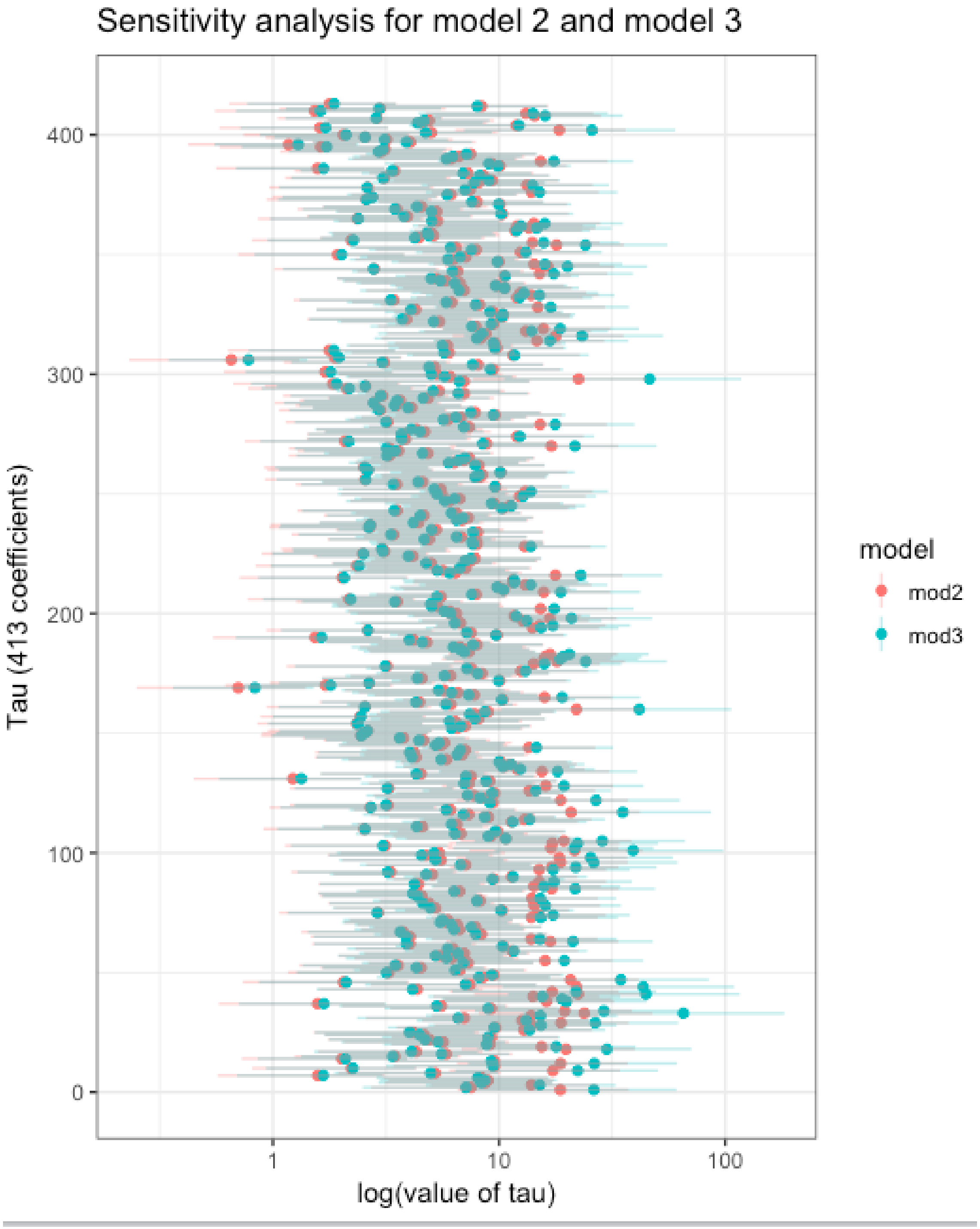

Time series
tau[1] chains 1:2
200.0
150.0
100.0
50.0
0.0
501
2000
4000
6000
iteration
tau[2] chains 1:2
30.0
20.0
10.0
0.0
501
2000
4000
6000
iteration

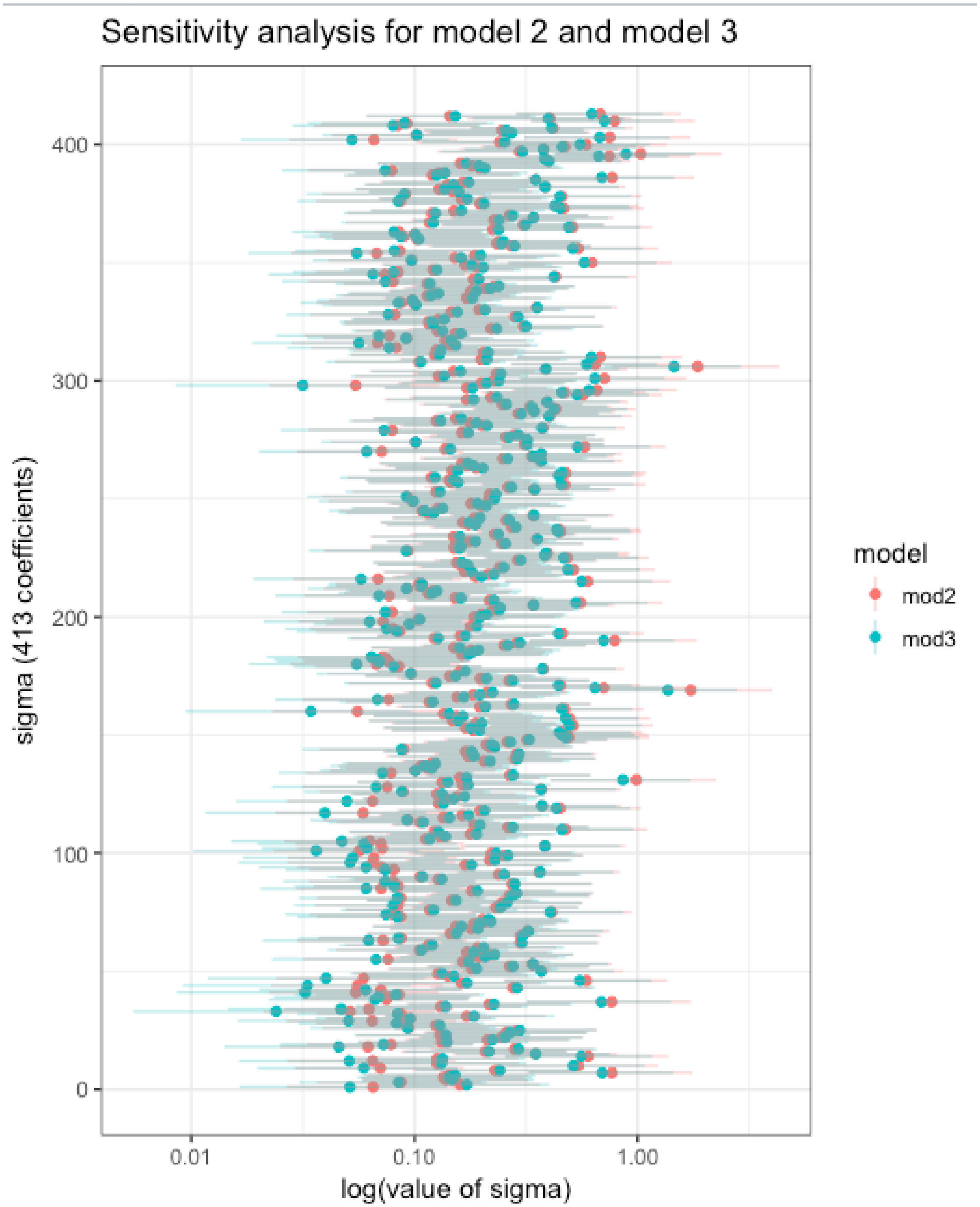

Autocorrelation function
tau[1] chains 1:2
tau[2] chains 1:2
tau[3] chains 1:2
tau[4] chains 1:2
tau[5] chains 1:2
tau[6] chains 1:2
tau[7] chains 1:2
tau[8] chains 1:2
tau[9] chains 1:2
tau[10] chains 1:2
tau[11] chains 1:2
tau[12] chains 1:2
1.0
0.5
0.0
-0.5
-1.0
0
20
40
lag


tau[7] chains 1:2
tau[8] chains 1:2
tau[9] chains 1:2
tau[10] chains 1:2
tau[11] chains 1:2
tau[12] chains 1:2
tau[13] chains 1:2
tau[14] chains 1:2
tau[15] chains 1:2
tau[16] chains 1:2
tau[17] chains 1:2
tau[18] chains 1:2
1.5
1.0
0.5
0.0
551
1000
2000
3000
start-iteration

Sensitivity to the prior of the parameters

To check the sensitivity with the election of a gamma prior distribution for σ_i or a lognormal, the mean and the 95% confidence interval for $i = 1, ..., 413$ is shown below for both σ and τ (the labels $\log(\sigma)$ and $\log(\tau)$ refers that the axis is in logarithm scale, not the values of σ and τ. This way, the values of tau are between $0 - 100$.)





We can see that the distributions of σ that have a mean close to 0 are a little bit different from model 2 to model 3 (and consequently the τ distribution is different). But, the values of the mean is different just by a small number, differing just in .01 or .02. Thus, we can assume that the selection of the prior has no important effect on the posterior distribution for τ and σ

Best model: DIC, posterior predictive checks and mixed predictive checks θ

For the posterior predictive checks and the mixed predictive checks, I will follow the ideas presented in the slides:

Posterior predictive checks (From the slides) At each iteration $j = 1, ..., N$ of the MCMC sampler:

- Generate $y_{grj}^{(pred)}$ from the full conditional for $y_{gr}^{(pred)}$
- Calculate $S_{gj}^{2(pred)} = \frac{\sum_r (y_{grj}^{(pred)} - y_{g \cdot j}^{(pred)})^2}{9}$
- Calculate $M_{gj} = I[S_{gj}^{2(pred)} \geq S_{gj}^{2(obs)}]$

The bayesian p-value for unit g is $p_g = \frac{\sum_{j=1}^{413} M_{gj}}{413}$

Mixed predictive checks (From the slides) At each iteration $j = 1, ..., N$ of the MCMC sampler:

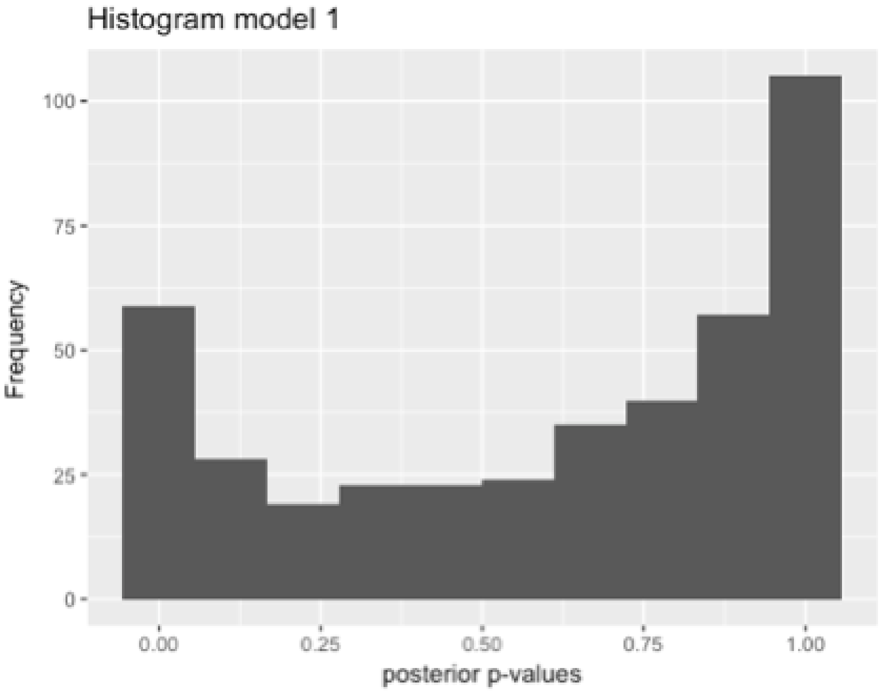
- Generate $sigma_{gj}^{(pred)}$ from the full conditional for $\sigma_g^{(pred)}$
- Generate $y_{grj}^{(pred)}$ from the full conditional for $y_{gr}^{(pred)}$ (using the σ_{gj}^{pred})
- Calculate $S_{gj}^{2(pred)} = \frac{\sum_r (y_{grj}^{(pred)} - y_{g \cdot j}^{(pred)})^2}{9}$
- Calculate $M_{gj} = I[S_{gj}^{2(pred)} \geq S_{gj}^{2(obs)}]$

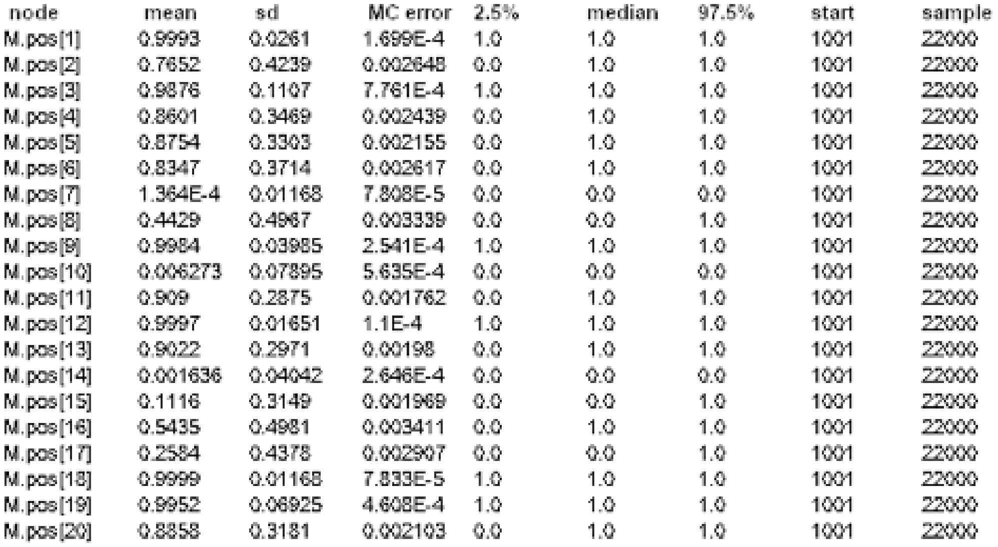
The bayesian p-value for unit g is $p_g = \frac{\sum_{j=1}^{413} M_{gj}}{413}$

According to the theory, the p-values follow an uniform distribution if model is *true*.

Now, for the first model just the posterior predictive checks will be made (no hyperpriors were used)

First model

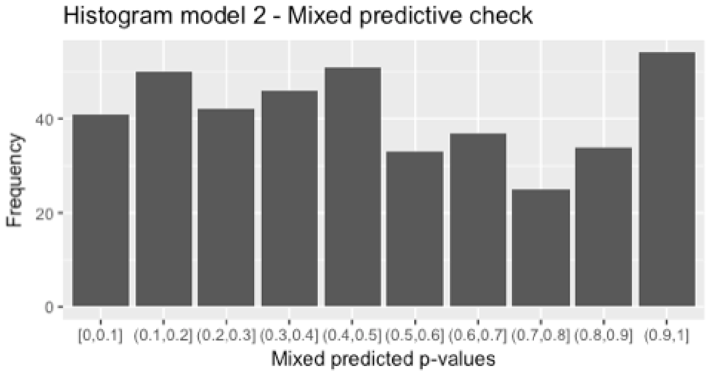


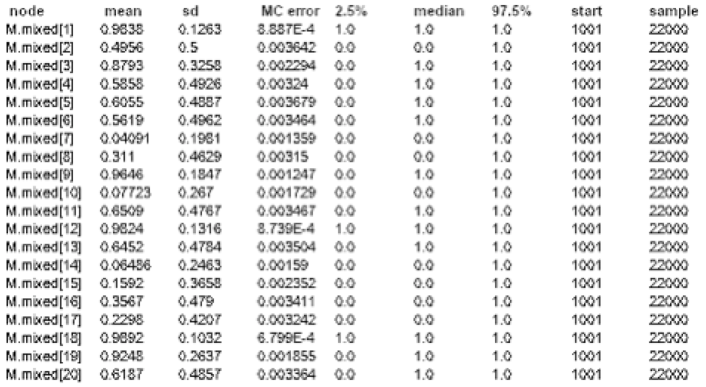


```
node        mean       sd        MC error  2.5%      median   97.5%    start     sample
M.pos[1]    0.9993     0.0261    1.699E-4  1.0       1.0      1.0      1001      22000
M.pos[2]    0.7652     0.4239    0.002648  0.0       1.0      1.0      1001      22000
M.pos[3]    0.9876     0.1107    7.761E-4  1.0       1.0      1.0      1001      22000
M.pos[4]    0.8601     0.3469    0.002439  0.0       1.0      1.0      1001      22000
M.pos[5]    0.8754     0.3303    0.002155  0.0       1.0      1.0      1001      22000
M.pos[6]    0.8347     0.3714    0.002617  0.0       1.0      1.0      1001      22000
M.pos[7]    1.364E-4   0.01168   7.808E-5  0.0       0.0      0.0      1001      22000
M.pos[8]    0.4429     0.4967    0.003339  0.0       0.0      1.0      1001      22000
M.pos[9]    0.9984     0.03985   2.541E-4  1.0       1.0      1.0      1001      22000
M.pos[10]   0.006273   0.07895   5.635E-4  0.0       0.0      0.0      1001      22000
M.pos[11]   0.909      0.2875    0.001762  0.0       1.0      1.0      1001      22000
M.pos[12]   0.9997     0.01651   1.1E-4    1.0       1.0      1.0      1001      22000
M.pos[13]   0.9022     0.2971    0.00198   0.0       1.0      1.0      1001      22000
M.pos[14]   0.001636   0.04042   2.646E-4  0.0       0.0      0.0      1001      22000
M.pos[15]   0.1116     0.3149    0.001969  0.0       0.0      1.0      1001      22000
M.pos[16]   0.5435     0.4981    0.003411  0.0       1.0      1.0      1001      22000
M.pos[17]   0.2584     0.4378    0.002907  0.0       0.0      1.0      1001      22000
M.pos[18]   0.9999     0.01168   7.833E-5  1.0       1.0      1.0      1001      22000
M.pos[19]   0.9952     0.06925   4.608E-4  1.0       1.0      1.0      1001      22000
M.pos[20]   0.8858     0.3181    0.002103  0.0       1.0      1.0      1001      22000
```

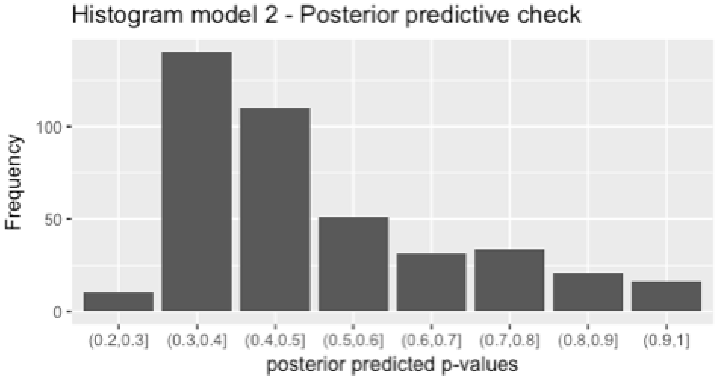
We can see that clearly, this does not follow an uniform distribution. For the second and third model, posterior predictive checks and mixed predictive checks will be made (as the methods in the lecture). As we can see, the plots related to the mixed predictive checks follows a more uniform distribution that the posterior predictive checks. Although is not perflectly uniform, the shape is very similar considering that we only have 413 elements.

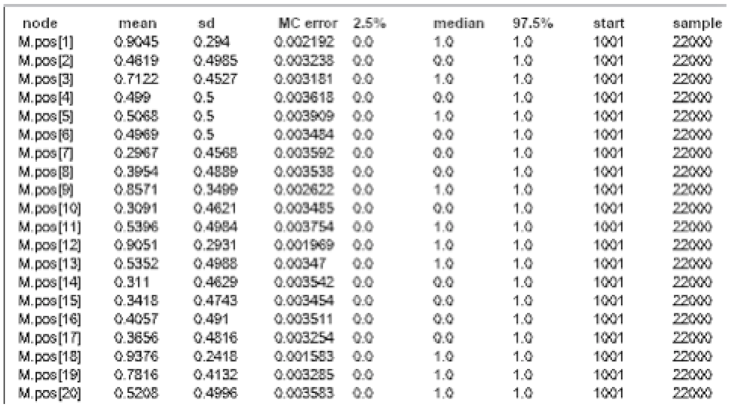
Second model





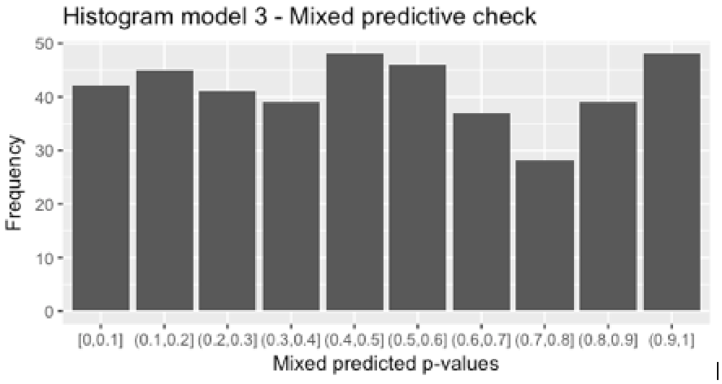
node	mean	sd	MC error	2.5%	median	97.5%	start	sample
M.mixed[1]	0.9838	0.1263	8.887E-4	1.0	1.0	1.0	1001	22000
M.mixed[2]	0.4956	0.5	0.003642	0.0	0.0	1.0	1001	22000
M.mixed[3]	0.8793	0.3258	0.002294	0.0	1.0	1.0	1001	22000
M.mixed[4]	0.5858	0.4926	0.00324	0.0	1.0	1.0	1001	22000
M.mixed[5]	0.6055	0.4887	0.003679	0.0	1.0	1.0	1001	22000
M.mixed[6]	0.5619	0.4962	0.003464	0.0	1.0	1.0	1001	22000
M.mixed[7]	0.04091	0.1981	0.001359	0.0	0.0	1.0	1001	22000
M.mixed[8]	0.311	0.4629	0.00315	0.0	0.0	1.0	1001	22000
M.mixed[9]	0.9646	0.1847	0.001247	0.0	1.0	1.0	1001	22000
M.mixed[10]	0.07723	0.267	0.001729	0.0	0.0	1.0	1001	22000
M.mixed[11]	0.6509	0.4767	0.003467	0.0	1.0	1.0	1001	22000
M.mixed[12]	0.9824	0.1316	8.739E-4	1.0	1.0	1.0	1001	22000
M.mixed[13]	0.6452	0.4784	0.003504	0.0	1.0	1.0	1001	22000
M.mixed[14]	0.06486	0.2463	0.00159	0.0	0.0	1.0	1001	22000
M.mixed[15]	0.1592	0.3658	0.002352	0.0	0.0	1.0	1001	22000
M.mixed[16]	0.3567	0.479	0.003411	0.0	0.0	1.0	1001	22000
M.mixed[17]	0.2298	0.4207	0.003242	0.0	0.0	1.0	1001	22000
M.mixed[18]	0.9892	0.1032	6.799E-4	1.0	1.0	1.0	1001	22000
M.mixed[19]	0.9248	0.2637	0.001855	0.0	1.0	1.0	1001	22000
M.mixed[20]	0.6187	0.4857	0.003364	0.0	1.0	1.0	1001	22000



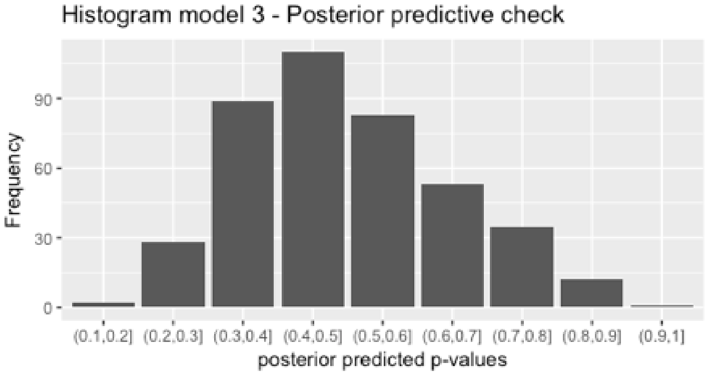


node	mean	sd	MC error	2.5%	median	97.5%	start	sample
M.pos[1]	0.9045	0.294	0.002192	0.0	1.0	1.0	1001	22000
M.pos[2]	0.4619	0.4985	0.003238	0.0	0.0	1.0	1001	22000
M.pos[3]	0.7122	0.4527	0.003181	0.0	1.0	1.0	1001	22000
M.pos[4]	0.499	0.5	0.003618	0.0	0.0	1.0	1001	22000
M.pos[5]	0.5068	0.5	0.003909	0.0	1.0	1.0	1001	22000
M.pos[6]	0.4969	0.5	0.003484	0.0	0.0	1.0	1001	22000
M.pos[7]	0.2967	0.4568	0.003592	0.0	0.0	1.0	1001	22000
M.pos[8]	0.3954	0.4889	0.003538	0.0	0.0	1.0	1001	22000
M.pos[9]	0.8571	0.3499	0.002622	0.0	1.0	1.0	1001	22000
M.pos[10]	0.3091	0.4621	0.003485	0.0	0.0	1.0	1001	22000
M.pos[11]	0.5396	0.4984	0.003754	0.0	1.0	1.0	1001	22000
M.pos[12]	0.9051	0.2931	0.001969	0.0	1.0	1.0	1001	22000
M.pos[13]	0.5352	0.4988	0.00347	0.0	1.0	1.0	1001	22000
M.pos[14]	0.311	0.4629	0.003542	0.0	0.0	1.0	1001	22000
M.pos[15]	0.3418	0.4743	0.003454	0.0	0.0	1.0	1001	22000
M.pos[16]	0.4057	0.491	0.003511	0.0	0.0	1.0	1001	22000
M.pos[17]	0.3656	0.4816	0.003254	0.0	0.0	1.0	1001	22000
M.pos[18]	0.9376	0.2418	0.001583	0.0	1.0	1.0	1001	22000
M.pos[19]	0.7816	0.4132	0.003285	0.0	1.0	1.0	1001	22000
M.pos[20]	0.5208	0.4996	0.003583	0.0	1.0	1.0	1001	22000

Third model



node	mean	sd	MC error	2.5%	median	97.5%	start	sample
M.mixed[1]	0.971	0.1678	0.001102	0.0	1.0	1.0	4001	22000
M.mixed[2]	0.5125	0.4998	0.003438	0.0	1.0	1.0	4001	22000
M.mixed[3]	0.8581	0.3489	0.00234	0.0	1.0	1.0	4001	22000
M.mixed[4]	0.5983	0.4902	0.003399	0.0	1.0	1.0	4001	22000
M.mixed[5]	0.62	0.4854	0.003652	0.0	1.0	1.0	4001	22000
M.mixed[6]	0.5825	0.4932	0.003432	0.0	1.0	1.0	4001	22000
M.mixed[7]	0.03364	0.1803	0.00128	0.0	0.0	1.0	4001	22000
M.mixed[8]	0.334	0.4716	0.003059	0.0	0.0	1.0	4001	22000
M.mixed[9]	0.9479	0.2222	0.001502	0.0	1.0	1.0	4001	22000
M.mixed[10]	0.07309	0.2603	0.001753	0.0	0.0	1.0	4001	22000
M.mixed[11]	0.653	0.476	0.003367	0.0	1.0	1.0	4001	22000
M.mixed[12]	0.9695	0.1721	0.001166	0.0	1.0	1.0	4001	22000
M.mixed[13]	0.6505	0.4768	0.003328	0.0	1.0	1.0	4001	22000
M.mixed[14]	0.05709	0.232	0.001586	0.0	0.0	1.0	4001	22000
M.mixed[15]	0.1726	0.3779	0.002684	0.0	0.0	1.0	4001	22000
M.mixed[16]	0.382	0.4859	0.003145	0.0	0.0	1.0	4001	22000
M.mixed[17]	0.2449	0.43	0.003182	0.0	0.0	1.0	4001	22000
M.mixed[18]	0.9803	0.1391	9.242E-4	1.0	1.0	1.0	4001	22000
M.mixed[19]	0.9063	0.2914	0.001972	0.0	1.0	1.0	4001	22000
M.mixed[20]	0.6261	0.4838	0.003364	0.0	1.0	1.0	4001	22000



node	mean	sd	MC error	2.5%	median	97.5%	start	sample
M.pos[1]	0.8011	0.3991	0.003949	0.0	1.0	1.0	4001	22000
M.pos[2]	0.4976	0.5	0.005025	0.0	0.0	1.0	4001	22000
M.pos[3]	0.6639	0.4724	0.004991	0.0	1.0	1.0	4001	22000
M.pos[4]	0.531	0.499	0.005111	0.0	1.0	1.0	4001	22000
M.pos[5]	0.548	0.4977	0.004919	0.0	1.0	1.0	4001	22000
M.pos[6]	0.5195	0.4996	0.005142	0.0	1.0	1.0	4001	22000
M.pos[7]	0.244	0.4295	0.004135	0.0	0.0	1.0	4001	22000
M.pos[8]	0.4285	0.4949	0.005064	0.0	0.0	1.0	4001	22000
M.pos[9]	0.7617	0.4261	0.004216	0.0	1.0	1.0	4001	22000
M.pos[10]	0.2922	0.4548	0.005135	0.0	0.0	1.0	4001	22000
M.pos[11]	0.5615	0.4962	0.004931	0.0	1.0	1.0	4001	22000
M.pos[12]	0.8052	0.396	0.003766	0.0	1.0	1.0	4001	22000
M.pos[13]	0.5557	0.4969	0.004753	0.0	1.0	1.0	4001	22000
M.pos[14]	0.2711	0.4445	0.004191	0.0	0.0	1.0	4001	22000
M.pos[15]	0.3618	0.4805	0.004762	0.0	0.0	1.0	4001	22000
M.pos[16]	0.452	0.4977	0.004751	0.0	0.0	1.0	4001	22000
M.pos[17]	0.3942	0.4887	0.005103	0.0	0.0	1.0	4001	22000
M.pos[18]	0.8256	0.3794	0.003833	0.0	1.0	1.0	4001	22000
M.pos[19]	0.71	0.4537	0.004309	0.0	1.0	1.0	4001	22000
M.pos[20]	0.5483	0.4977	0.005468	0.0	1.0	1.0	4001	22000

Lastly, the DIC from the three models will be compared

Table 4: DIC comparison

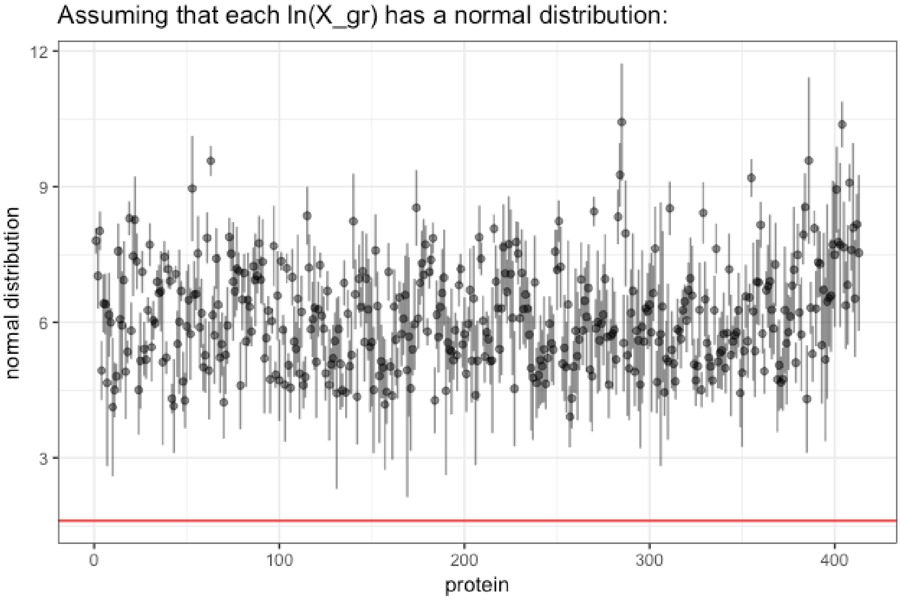
Model	DIC
Model 1: Equal variance for all proteins	5411.4
Model 2: Hierarchical (gamma)	4565.41
Model 3: Hierarchical (log-normal)	4476.52

From this table we can see that the best of the three models is the Hierarchical (log-normal).

3 Missing values

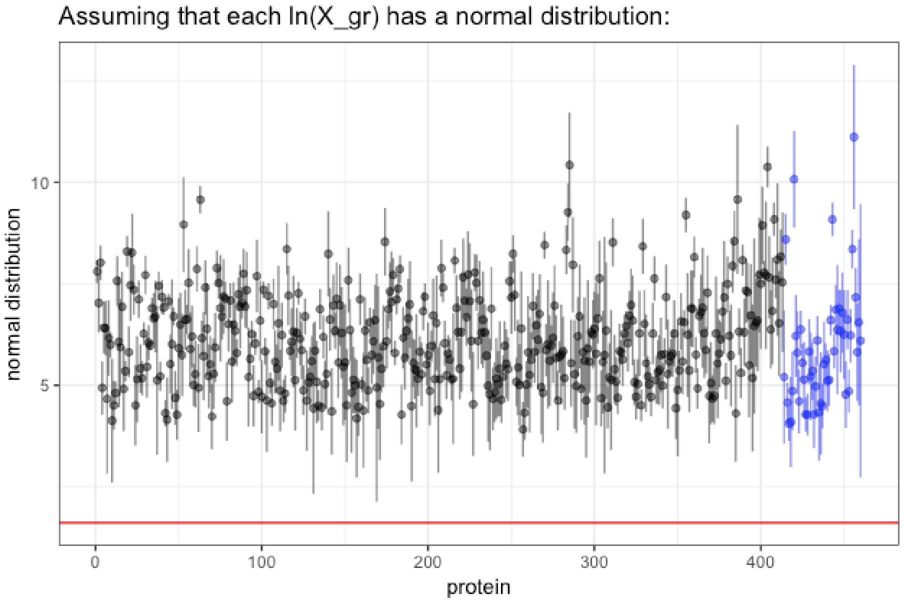
Missing at random assumption (MAR)

At first sigth, the missing data sounds like a censored data problem [1], because the data that is less than 5 is missing, so the missigness depend on the value of the dependant variable. But, analyzing deeply the data, we can see that, for each protein **is not common** that values below 5 appear in the dataset (in the plot below is log(5) because $log(X_{gr}) \sim N(\mu_g, \sigma_g^2))$:



Now, the data does not look like a censored data problem. For me, it just seems as a method to eliminate noisy data that arises just because of external factors (or maybe just corrupted data). Thus, I will work with this problem with an *ignorable missing data mechanism.* Another question that we have to ask is if the proteins with missing data are completely different than the proteins with no missing data (This to express the idea that the number of missign values is in function of the protein itself):

[1] like the problem that arises when asking earning to people. People that earns too much money does not answer the question



In the above plot we can see that the empirical distribution of the data with missing values (in blue) is quite similar to the one that have complete data. There is no evidence to assume that the distributions are different (and in consequence a *reason* to have missing values).

Some extra information that can be useful to analyze in order to know if the missing data has some estructure is the number of missing values per rat:

Table 5: Number of missing values per rat

Rat number	Missing values (MV)	(MV)/47	(MV)/460
1	0	0 %	0 %
2	6	12.8 %	1.3%
3	25	53.2 %	5.4%
4	5	10.6%	1.1%
5	8	17.0%	1.7%
6	12	25.5%	2.6%
7	4	8.5%	0.9%
8	11	23.4%	2.4%
9	9	19.1%	2.0%

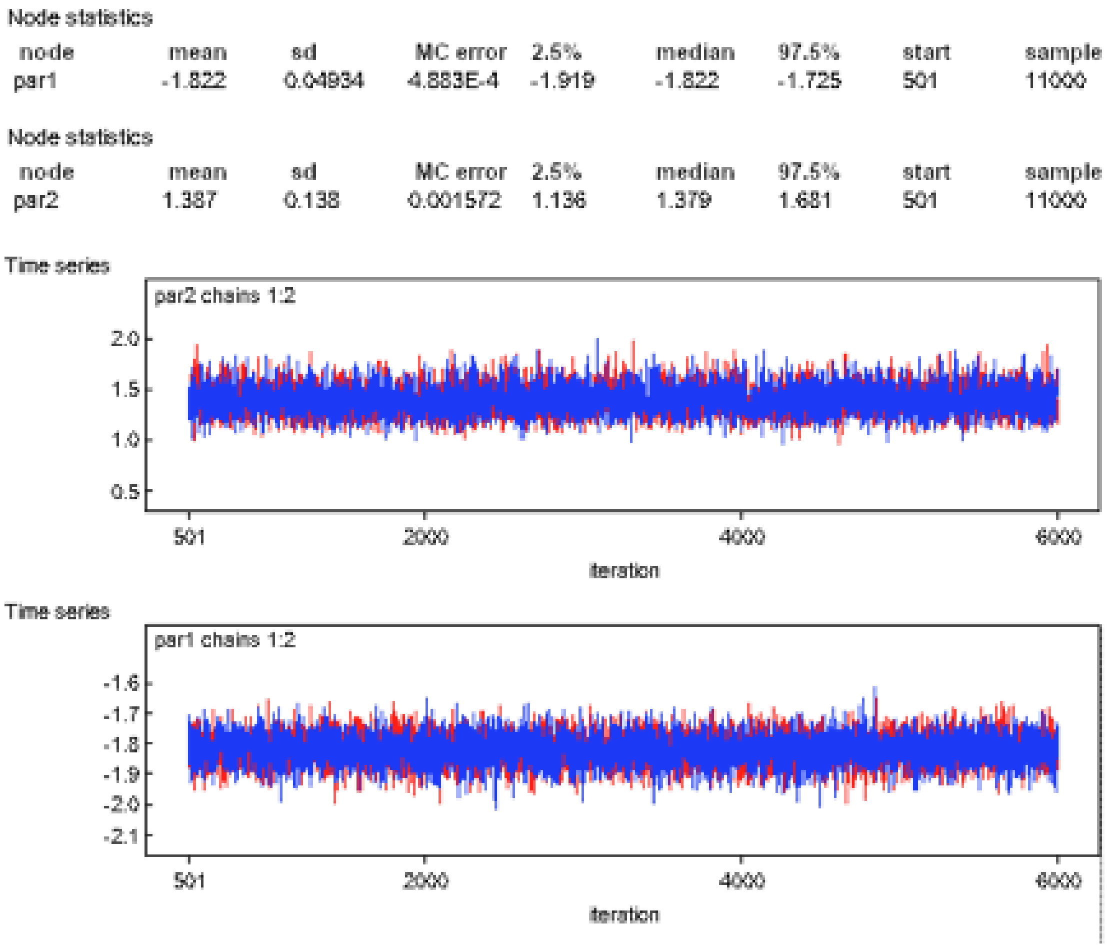
The rat 3 has 25 missing values (that too much compared with the rat 1!). This show us that the missing values are not sufficiently at random, but we do not have sufficient evidence (it can be just by chance, the range are just from 0% to 5.4%). To make a conclussion about the relation rat number - number of missing values we would need more information or an explanation of this factor.

As a conclusion and for the rest of this report, I will assume that the data is missing at random.

Models for missing data

For this section, the posterior distribution of the hyperparameters under the best model in section 1 is used as a prior distribution. I found impossible to use the exact posterior distribution found in section 1. As suggested, I will use the normal and gamma distributions that approximate these posteriors. (Although these do not completely especify the posterior distribution in the best model of section 1)

For each μ_i a Normal non-informative distribution was used $N(0, 1000)$. But, for the each τ_i a lognormal distribution was used with parameters par_1, par_2, with prior for $par_1 \sim N(\mu_{par_1}, \sigma_{par_1})$ and $par_2 \sim gamma(alpha_{par_2}, beta_{par_2})$. In the posterior summaries from section one, the mean and sd for each of these parameters are the next:



Node statistics

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
par1	-1.822	0.04934	4.883E-4	-1.919	-1.822	-1.725	501	11000

Node statistics

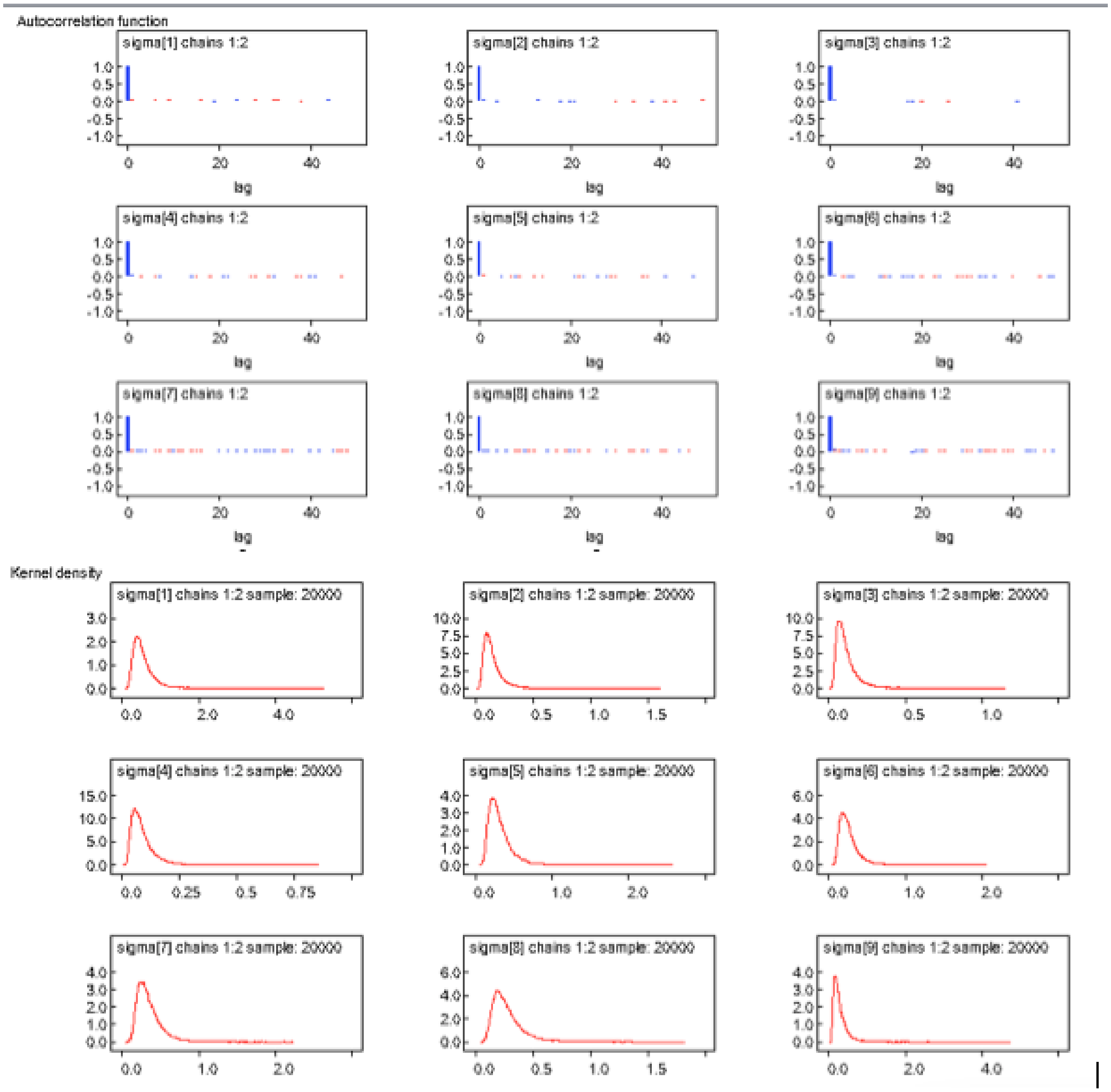
node	mean	sd	MC error	2.5%	median	97.5%	start	sample
par2	1.387	0.138	0.001572	1.136	1.379	1.681	501	11000

Then, par_1 can be approximated with a normal distribution $Normal(-1.822, 0.0024)$ (In terms of mean and variance). Equivalently, par_2 can be approximated with a lognormal distribution $lognormal(101.02, 72.83)$

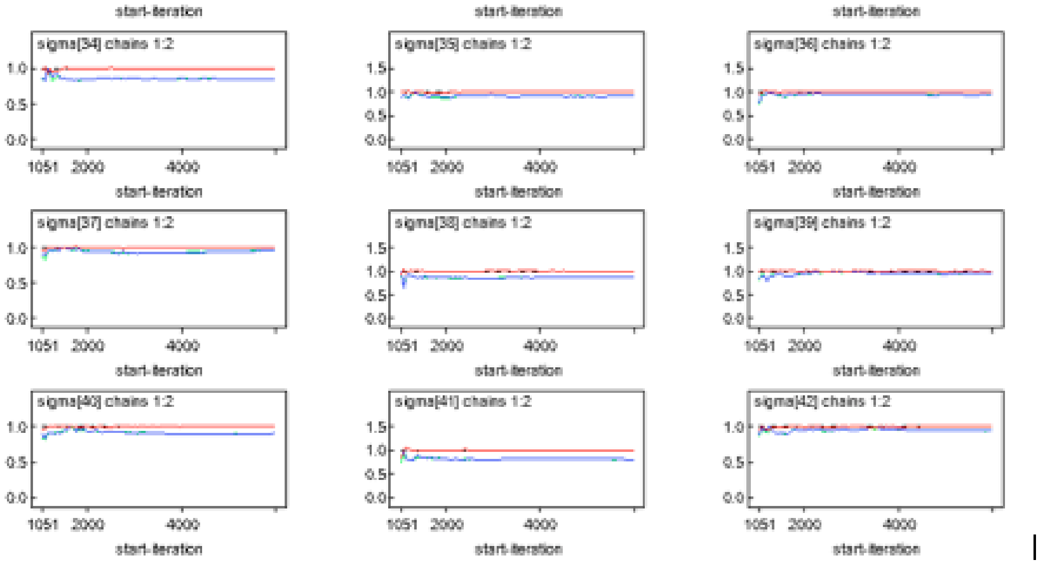
Now, the analysis for missing data will be made in two ways. In the first, the missing data will be ignored (model 4). In the second, the missing data will be imputed (model 5). Because of the objective of the report, and to not make it excesivelly long, just the parameters σ_i and τ_i will be analyzed

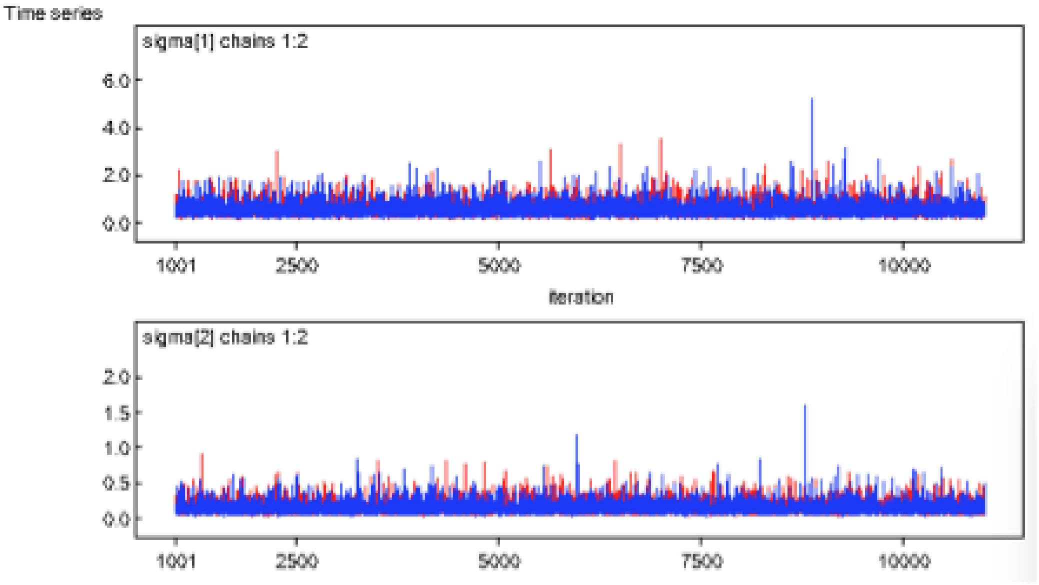
Ignoring missing data (model 4)

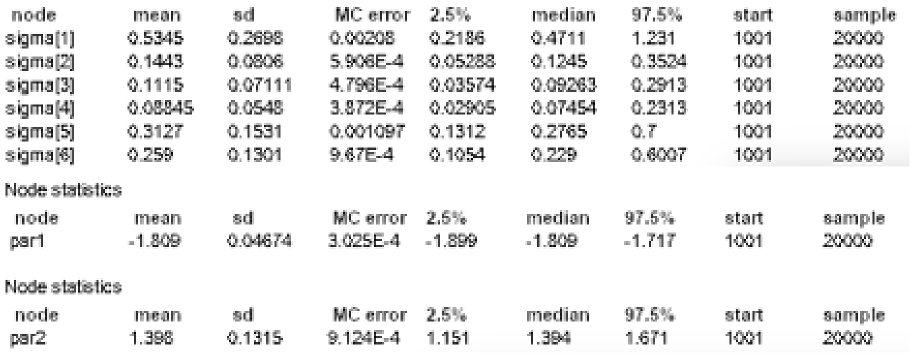
In this subsection, the used data file was *DataProteinsMissing.txt*. 11,000 iterations were made with a thinning of 10 and a burning of 1001. This datafile does not contains the missing values as NA. First, the autocorrelation and the density of the posterior distribution of the first 9 σ_i will be analyzed:



Now, the bgr statistic and the chains will be analyzed. From the first we can see that the convergence looks good. For the second, we can observe that the chain looks good too.







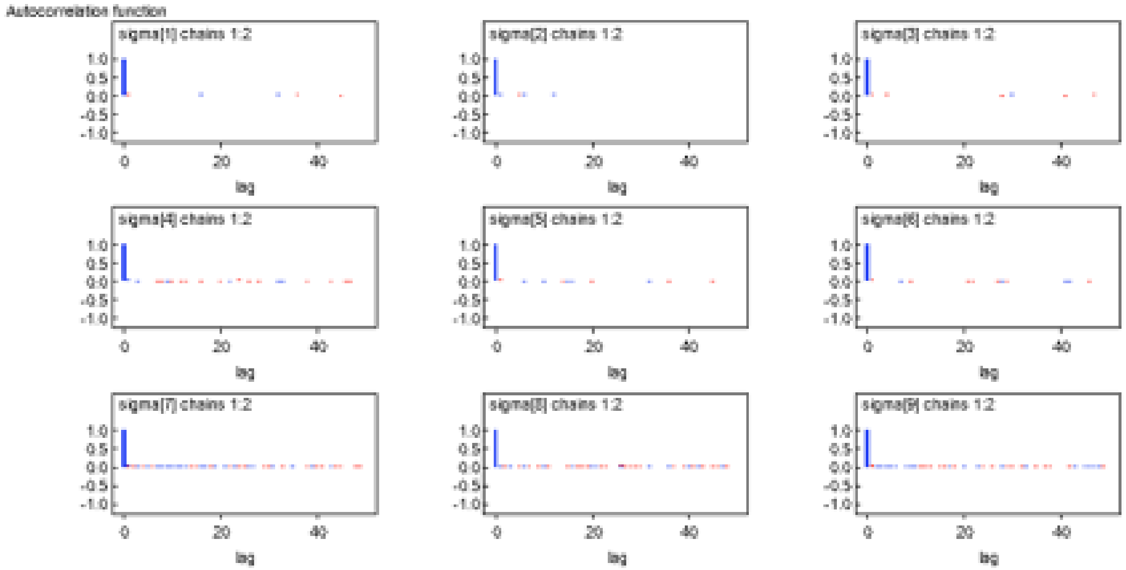
```
node      mean     sd       MC error   2.5%     median   97.5%   start  sample
sigma[1]  0.5345   0.2698   0.00208    0.2186   0.4711   1.231   1001   20000
sigma[2]  0.1443   0.0806   5.906E-4   0.05288  0.1245   0.3524  1001   20000
sigma[3]  0.1115   0.07111  4.796E-4   0.03574  0.09263  0.2913  1001   20000
sigma[4]  0.08845  0.0548   3.872E-4   0.02905  0.07454  0.2313  1001   20000
sigma[5]  0.3127   0.1531   0.001097   0.1312   0.2765   0.7     1001   20000
sigma[6]  0.259    0.1301   9.67E-4    0.1054   0.229    0.6007  1001   20000
```

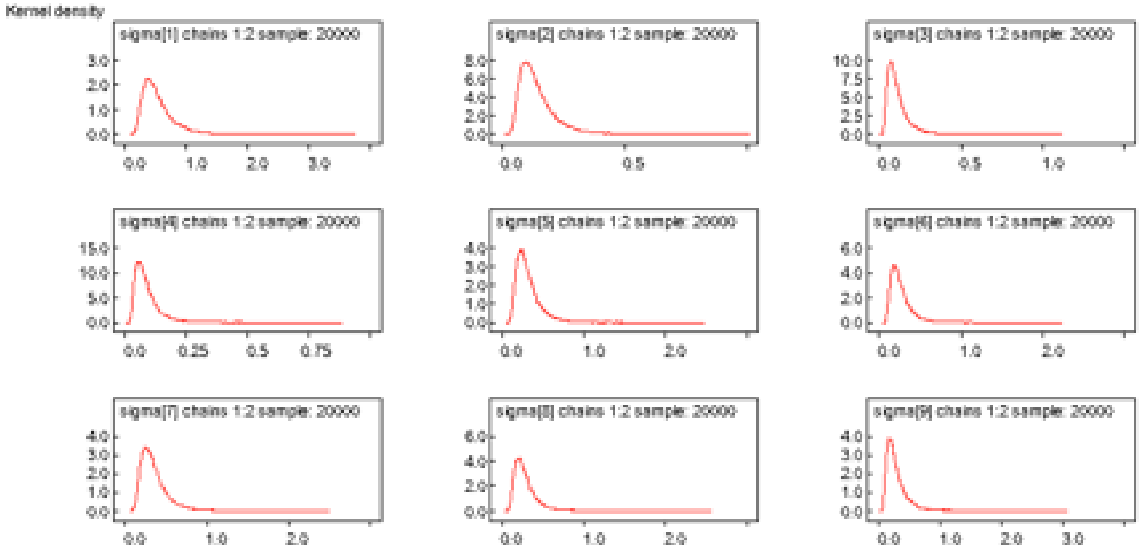
```
Node statistics
 node     mean     sd       MC error   2.5%     median   97.5%   start  sample
 par1     -1.809   0.04674  3.025E-4   -1.899   -1.809   -1.717  1001   20000
```

```
Node statistics
 node     mean     sd       MC error   2.5%     median   97.5%   start  sample
 par2     1.398    0.1315   9.124E-4   1.151    1.394    1.671   1001   20000
```

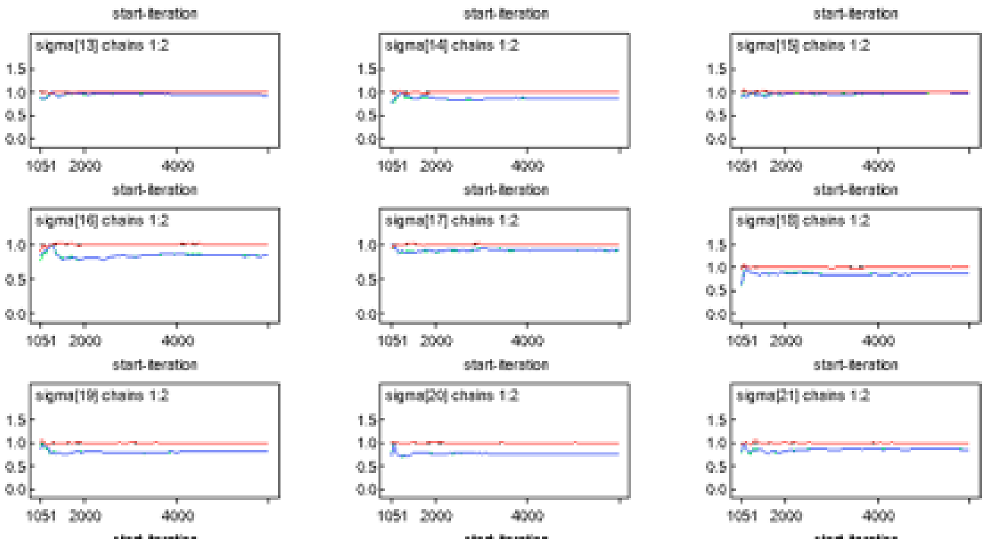
Imputing values (model 5)

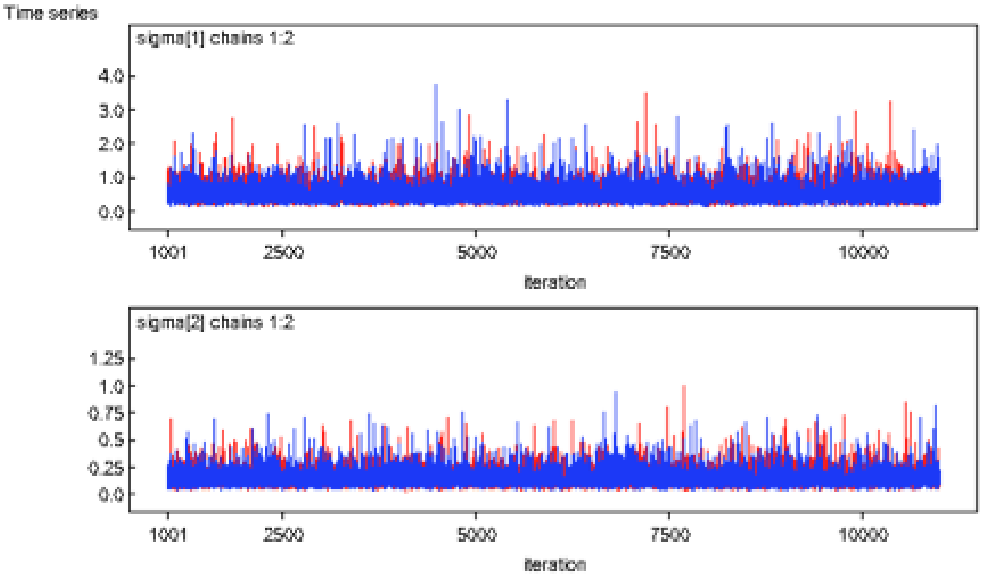
In this subsection, the data file used was *DataProteinsMissingNA.txt*. 11,000 iterations were made with a thinning of 10 and a burning of 1001. This datafile contains the missing values as NA. First the autocorrelation and the density of the posterior distribution of the first 9 σ_i will be analyzed:

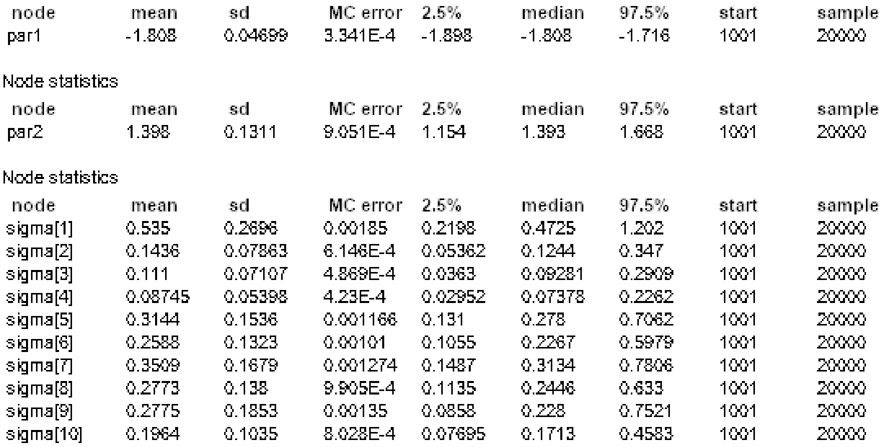




Now, the bgr statistic and the chains will be analyzed. From the first we can see that the convergence looks good. For the second, we can observe that the chain looks good too.







node	mean	sd	MC error	2.5%	median	97.5%	start	sample
par1	-1.808	0.04699	3.341E-4	-1.898	-1.808	-1.716	1001	20000

Node statistics

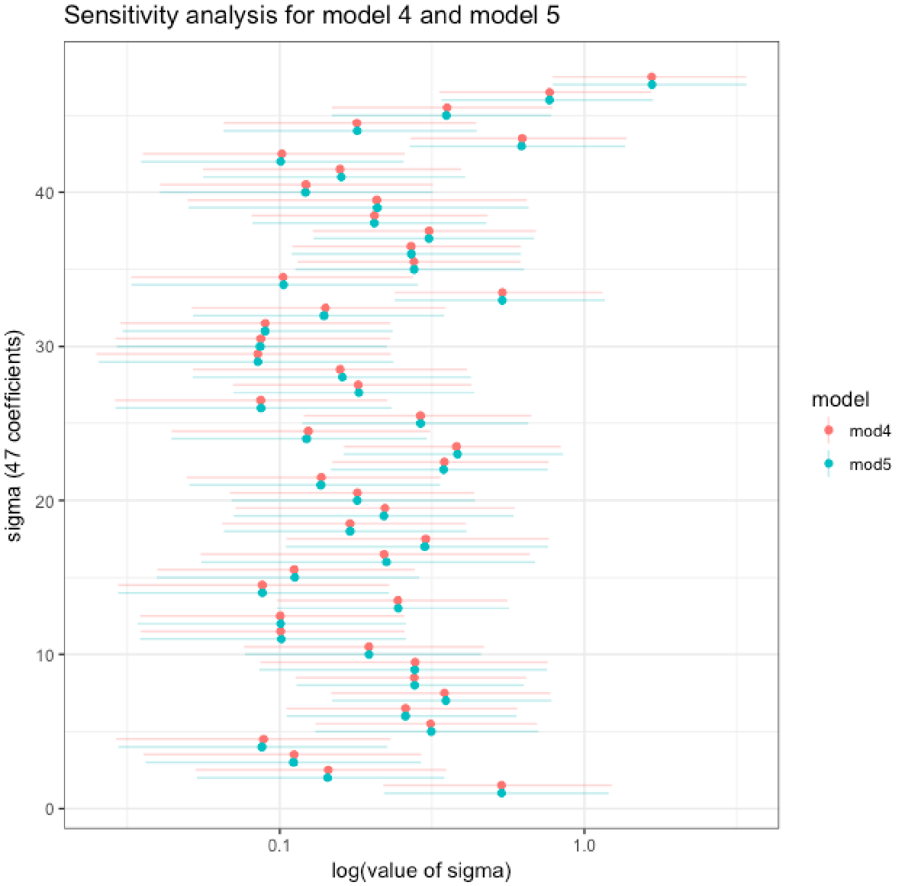
node	mean	sd	MC error	2.5%	median	97.5%	start	sample
par2	1.398	0.1311	9.051E-4	1.154	1.393	1.668	1001	20000

Node statistics

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
sigma[1]	0.535	0.2696	0.00185	0.2198	0.4725	1.202	1001	20000
sigma[2]	0.1436	0.07863	6.146E-4	0.05362	0.1244	0.347	1001	20000
sigma[3]	0.111	0.07107	4.869E-4	0.0363	0.09281	0.2909	1001	20000
sigma[4]	0.08745	0.05398	4.23E-4	0.02952	0.07378	0.2262	1001	20000
sigma[5]	0.3144	0.1536	0.001166	0.131	0.278	0.7062	1001	20000
sigma[6]	0.2588	0.1323	0.00101	0.1055	0.2267	0.5979	1001	20000
sigma[7]	0.3509	0.1679	0.001274	0.1487	0.3134	0.7806	1001	20000
sigma[8]	0.2773	0.138	9.905E-4	0.1135	0.2446	0.633	1001	20000
sigma[9]	0.2775	0.1853	0.00135	0.0858	0.228	0.7521	1001	20000
sigma[10]	0.1964	0.1035	8.028E-4	0.07695	0.1713	0.4583	1001	20000

Comparison model 4 and model 5

To compare the model 4 and model 5, the distributions of each σ_i are ploted together. In the next plot we can see how they look similar:



As the slides of the course say “Results with missing data same as complete-case analysis here, since missing values are only in response, and mechanism is assumed ignorable”. In this example we have only missing values in response and we assumed ignorable the mechanism, so the distributions look almost the same.

4 Conclusions

From the first section of the report, the hierarchical model (with lognormal prior) is the best one. When trying to model the missing data I assume that the mechanism is ignorable (although I found some evidence that some rats have more missing values than others, but the difference was not very big. Also, we do not have information to assume that some ”kind” of rats has more missing values, as the example in the slides of the fat rats). Some ideas to improve the models is to create variances for some groups of proteins (for example, make cluster of proteins and model just one variance for all of them).

5 Code for the models

Model 1

```
model
{
  for( i in 1 : p ) {
    for( j in 1:n){
      x[i,j] <- log(y[i,j])
      x[i,j] ~ dnorm(mu[i], tau)

      # Posterior predictions
      pred.pos[i,j] ~ dnorm(mu[i], tau)
    }
  mean.orig[i]<- (x[i,1]+x[i,2]+x[i,3]+x[i,4]+x[i,5]+x[i,6]+x[i,7]+x[i,8]+x[i ,9]) /9
  s_2.orig[i]<-(pow(x[i,1]-mean.orig[i], 2)+pow(x[i,2]-mean.orig[i], 2)+pow(x[i,3]-mean.orig[i],
     2)+pow(x[i,4]-mean.orig[i], 2)+pow(x[i,5]-mean.orig[i], 2)+pow(x[i,6]-mean.orig[i],
     2)+pow(x[i,7]-mean.orig[i], 2)+pow(x[i,8]-mean.orig[i], 2)+pow(x[i,9]-mean.orig[i], 2))/9

  mean.pos[i] <-
     (pred.pos[i,1]+pred.pos[i,2]+pred.pos[i,3]+pred.pos[i,4]+pred.pos[i,5]+pred.pos[i,6]+pred.pos[i,7]+
  pred.pos[i,8]+pred.pos[i ,9]) /9

  s_2.pos[i] <- (pow(pred.pos[i,1]-mean.pos[i], 2)+pow(pred.pos[i,2]-mean.pos[i],
     2)+pow(pred.pos[i,3]-mean.pos[i], 2)+pow(pred.pos[i,4]-mean.pos[i],
     2)+pow(pred.pos[i,5]-mean.pos[i], 2)+pow(pred.pos[i,6]-mean.pos[i],
     2)+pow(pred.pos[i,7]-mean.pos[i], 2)+pow(pred.pos[i,8]-mean.pos[i],
     2)+pow(pred.pos[i,9]-mean.pos[i], 2))/9

  mu[i] ~ dnorm(0, .0001)
  M.pos[i] <- step(s_2.pos[i]-s_2.orig[i])
   diff[i] <- s_2.pos[i]-s_2.orig[i]
  }
  tau ~ dgamma(.01, .01)
  sigma <- 1/tau

}
```

Model 2

```
model
{
  for( i in 1 : p ) {
    for( j in 1:n){
      x[i,j] <- log(y[i,j])
      x[i,j] ~ dnorm(mu[i], tau[i])

      # Posterior predictions
      pred.pos[i,j] ~ dnorm(mu[i], tau[i])

      # Mixed predictions
      pred.mix[i,j] ~ dnorm(mu[i], tau.pred[i])
    }


  mean.orig[i]<- (x[i,1]+x[i,2]+x[i,3]+x[i,4]+x[i,5]+x[i,6]+x[i,7]+x[i,8]+x[i,9]) /9
  s_2.orig[i]<-(pow(x[i,1]-mean.orig[i], 2)+pow(x[i,2]-mean.orig[i], 2)+pow(x[i,3]-mean.orig[i],
     2)+pow(x[i,4]-mean.orig[i], 2)+pow(x[i,5]-mean.orig[i], 2)+pow(x[i,6]-mean.orig[i],
     2)+pow(x[i,7]-mean.orig[i], 2)+pow(x[i,8]-mean.orig[i], 2)+pow(x[i,9]-mean.orig[i], 2))/9

  mean.pos[i] <-
     (pred.pos[i,1]+pred.pos[i,2]+pred.pos[i,3]+pred.pos[i,4]+pred.pos[i,5]+pred.pos[i,6]+pred.pos[i,7]+
  pred.pos[i,8]+pred.pos[i,9]) /9

  s_2.pos[i] <- (pow(pred.pos[i,1]-mean.pos[i], 2)+pow(pred.pos[i,2]-mean.pos[i],
     2)+pow(pred.pos[i,3]-mean.pos[i], 2)+pow(pred.pos[i,4]-mean.pos[i],
     2)+pow(pred.pos[i,5]-mean.pos[i], 2)+pow(pred.pos[i,6]-mean.pos[i],
     2)+pow(pred.pos[i,7]-mean.pos[i], 2)+pow(pred.pos[i,8]-mean.pos[i],
     2)+pow(pred.pos[i,9]-mean.pos[i], 2))/9

  mean.mix[i] <-
     (pred.mix[i,1]+pred.mix[i,2]+pred.mix[i,3]+pred.mix[i,4]+pred.mix[i,5]+pred.mix[i,6]+pred.mix[i,7]+
  pred.mix[i,8]+pred.mix[i,9]) /9

  s_2.mix[i] <- (pow(pred.mix[i,1]-mean.mix[i], 2)+pow(pred.mix[i,2]-mean.mix[i],
     2)+pow(pred.mix[i,3]-mean.mix[i], 2)+pow(pred.mix[i,4]-mean.mix[i],
     2)+pow(pred.mix[i,5]−mean.mix[i], 2)+pow(pred.mix[i,6]−mean.mix[i],
     2)+pow(pred.mix[i,7]−mean.mix[i], 2)+pow(pred.mix[i,8]−mean.mix[i],
     2)+pow(pred.mix[i,9]−mean.mix[i], 2))/9

  mu[i] ~ dnorm(0, .0001)
  tau.pred[i] ~ dgamma(a, b) # for mixed predictions
  tau[i] ~ dgamma(a, b)
  sigma[i] <− 1/tau[i]
  M.pos[i] <− step(s_2.pos[i]−s_2.orig[i])
  M.mixed[i] <− step(s_2.mix[i]−s_2.orig[i])
  }
  a ~ dgamma(0.01, 0.01)
  b ~ dgamma(0.01, 0.01)
}
```

Model 3

```
model
{
   for( i in 1 : p ) {
      for( j in 1:n){
         x[i,j] <− log(y[i,j])
         x[i,j] ~ dnorm(mu[i], tau[i])

         # Posterior predictions
         pred.pos[i,j] ~ dnorm(mu[i], tau[i])

         # Mixed predictions
         pred.mix[i,j] ~ dnorm(mu[i], tau.pred[i])
       }

mean.orig[i]<− (x[i,1]+x[i,2]+x[i,3]+x[i,4]+x[i,5]+x[i,6]+x[i,7]+x[i,8]+x[i,9]) /9
   s_2.orig[i]<−(pow(x[i,1]−mean.orig[i], 2)+pow(x[i,2]−mean.orig[i], 2)+pow(x[i,3]−mean.orig[i],
        2)+pow(x[i,4]−mean.orig[i], 2)+pow(x[i,5]−mean.orig[i], 2)+pow(x[i,6]−mean.orig[i],
        2)+pow(x[i,7]−mean.orig[i], 2)+pow(x[i,8]−mean.orig[i], 2)+pow(x[i,9]−mean.orig[i], 2))/9
```

```
  mean.pos[i] <-
      (pred.pos[i,1]+pred.pos[i,2]+pred.pos[i,3]+pred.pos[i,4]+pred.pos[i,5]+pred.pos[i,6]+pred.pos[i,7]+
  pred.pos[i,8]+pred.pos[i ,9]) /9

  s_2.pos[i] <- (pow(pred.pos[i,1]-mean.pos[i], 2)+pow(pred.pos[i,2]-mean.pos[i],
      2)+pow(pred.pos[i,3]-mean.pos[i], 2)+pow(pred.pos[i,4]-mean.pos[i],
      2)+pow(pred.pos[i,5]-mean.pos[i], 2)+pow(pred.pos[i,6]-mean.pos[i],
      2)+pow(pred.pos[i,7]-mean.pos[i], 2)+pow(pred.pos[i,8]-mean.pos[i],
      2)+pow(pred.pos[i,9]-mean.pos[i], 2))/9

  mean.mix[i] <-
      (pred.mix[i,1]+pred.mix[i,2]+pred.mix[i,3]+pred.mix[i,4]+pred.mix[i,5]+pred.mix[i,6]+pred.mix[i,7]+
  pred.mix[i,8]+pred.mix[i ,9]) /9

  s_2.mix[i] <- (pow(pred.mix[i,1]-mean.mix[i], 2)+pow(pred.mix[i,2]-mean.mix[i],
      2)+pow(pred.mix[i,3]-mean.mix[i], 2)+pow(pred.mix[i,4]-mean.mix[i],
      2)+pow(pred.mix[i,5]-mean.mix[i], 2)+pow(pred.mix[i,6]-mean.mix[i],
      2)+pow(pred.mix[i,7]-mean.mix[i], 2)+pow(pred.mix[i,8]-mean.mix[i],
      2)+pow(pred.mix[i,9]-mean.mix[i], 2))/9



  mu[i] ~ dnorm(0, .0001)
  log_sigma[i] ~ dnorm(par1, par2)
  log_sigma.pred[i]~dnorm(par1, par2)

  tau[i] <- 1/exp(log_sigma[i])
  tau.pred[i] <- 1/exp(log_sigma.pred[i])

  sigma[i] <- 1/tau[i]
  M.pos[i]<-step(s_2.pos[i]-s_2.orig[i])
  M.mixed[i]<-step(s_2.mix[i]-s_2.orig[i])
  }
  par1 ~ dnorm(0, .001)
  par2 ~ dgamma(0.01, 0.01)
}
```

Model 4

```
model
{
   for( i in 1 : N ) {
      y[i] ~ dlnorm(mu[protein[i]], tau[protein[i]])
      }
   for( i in 1:p){
      mu[i] ~ dnorm(0, .0001)
      log_sigma[i] ~ dnorm(par1, par2)
      tau[i] <- 1/exp(log_sigma[i])
      sigma[i] <- 1/tau[i]
   }
   par1 ~ dnorm(-1.822, 410.77283)
   par2 ~ dgamma(101.02, 72.83)
}
```

Model 5

```
model
{
   for( i in 1 : p ) {
      for( j in 1:n){
      y[i,j] ~ dlnorm(mu[i], tau[i])
      }
   }
   for( i in 1:p){
      mu[i] ~ dnorm(0, .0001)
      log_sigma[i] ~ dnorm(par1, par2)
      tau[i] <- 1/exp(log_sigma[i])
      sigma[i] <- 1/tau[i]
   }
   par1 ~ dnorm(-1.822, 410.77283)
   par2 ~ dgamma(101.02, 72.83)
}
```