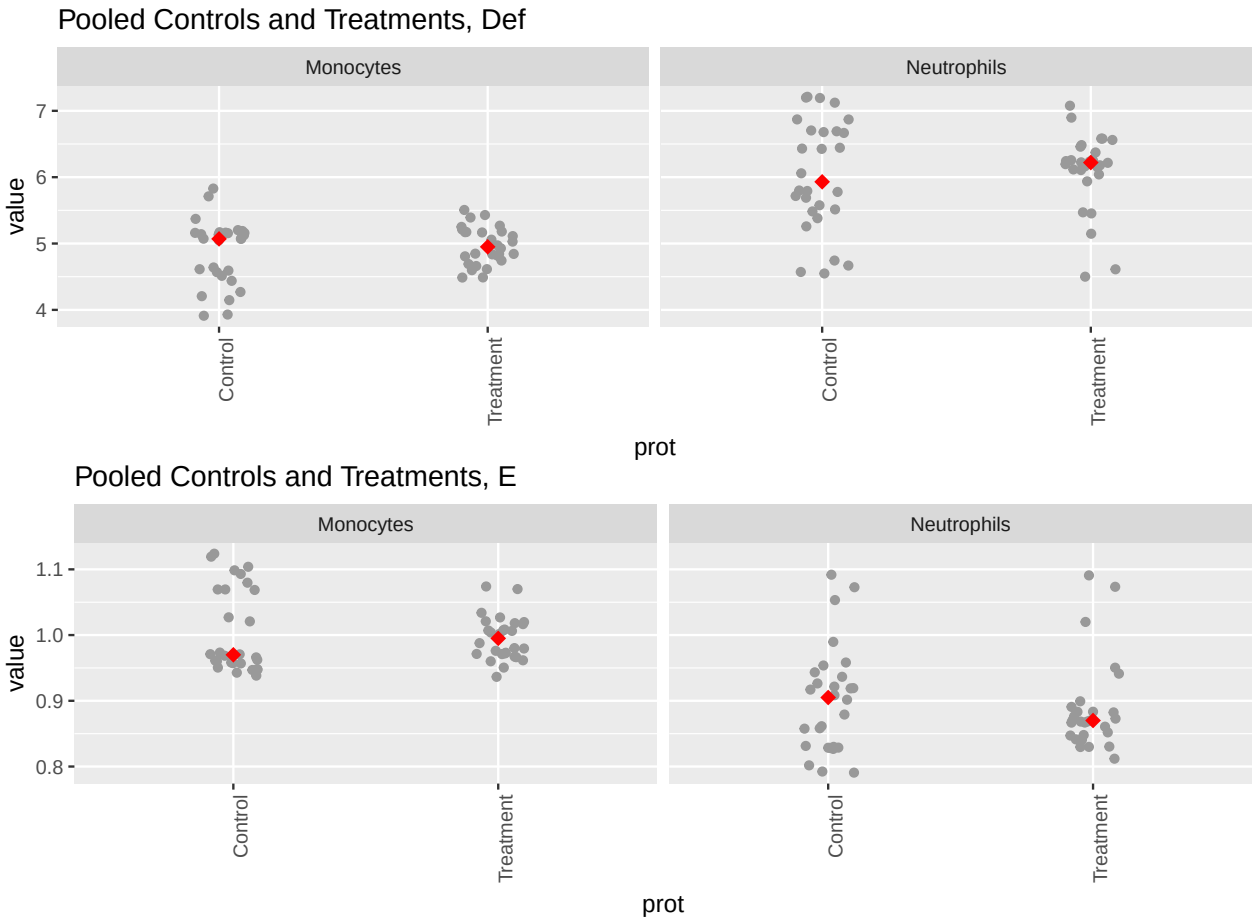
RT-DC Data

Setup

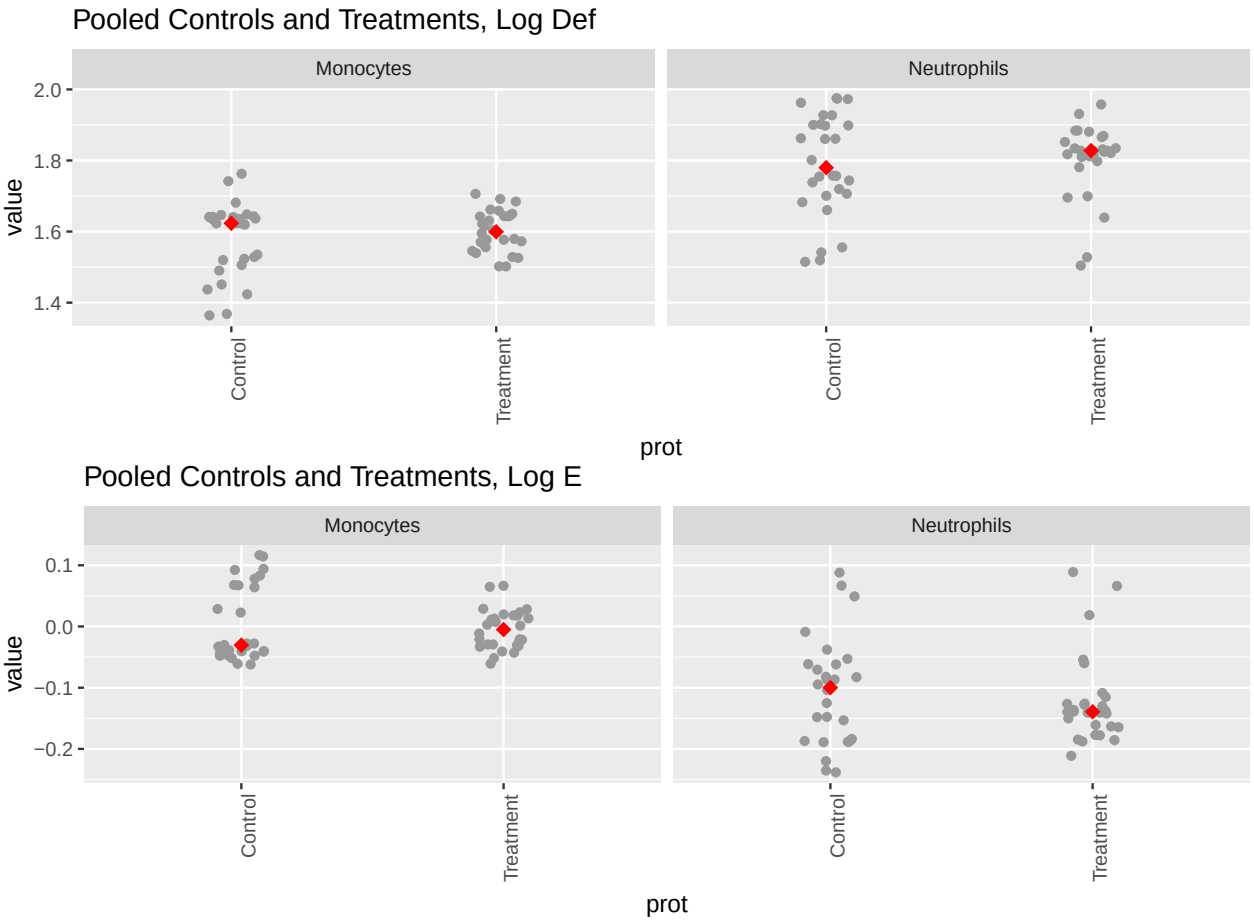
Load and clean data

```
source("load_rtdc_3.R")
XL_PATH <- file.path(PROJ_DIR, "rt-dc_eb_2.xls")
rtdc_tall <- load_rtdc_3(XL_PATH)
```

We evaluate whether data is a better fit to normal or log-normal. First view without taking the log:

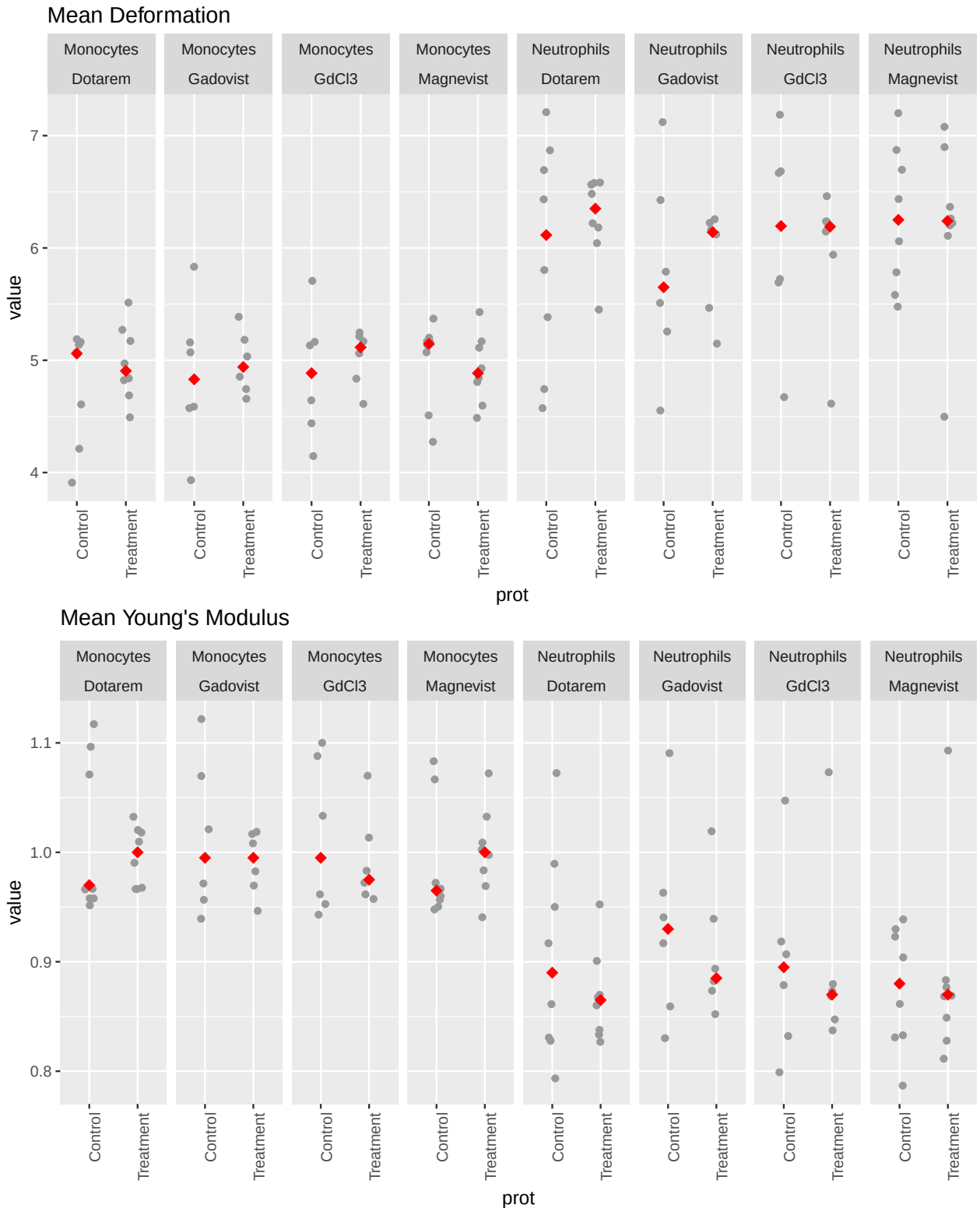


Deformation looks relatively normal but E is skewed right (towards the top of the chart).

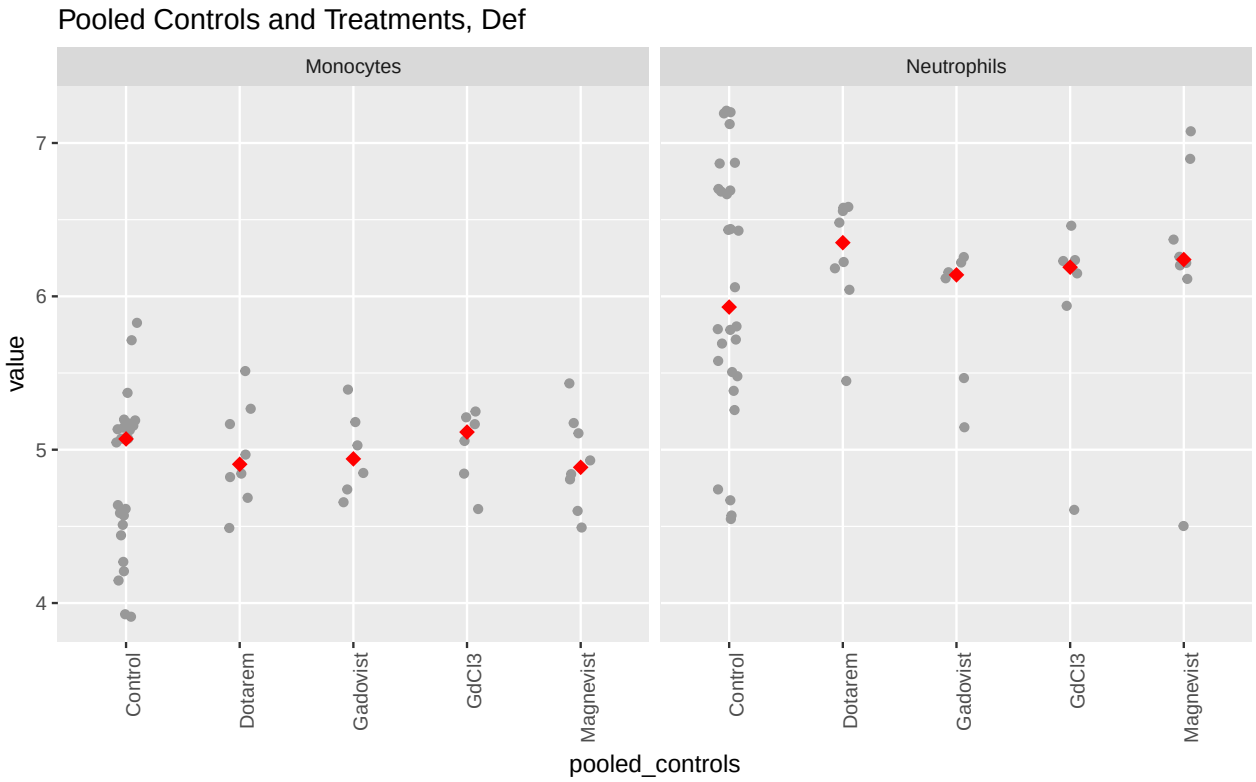


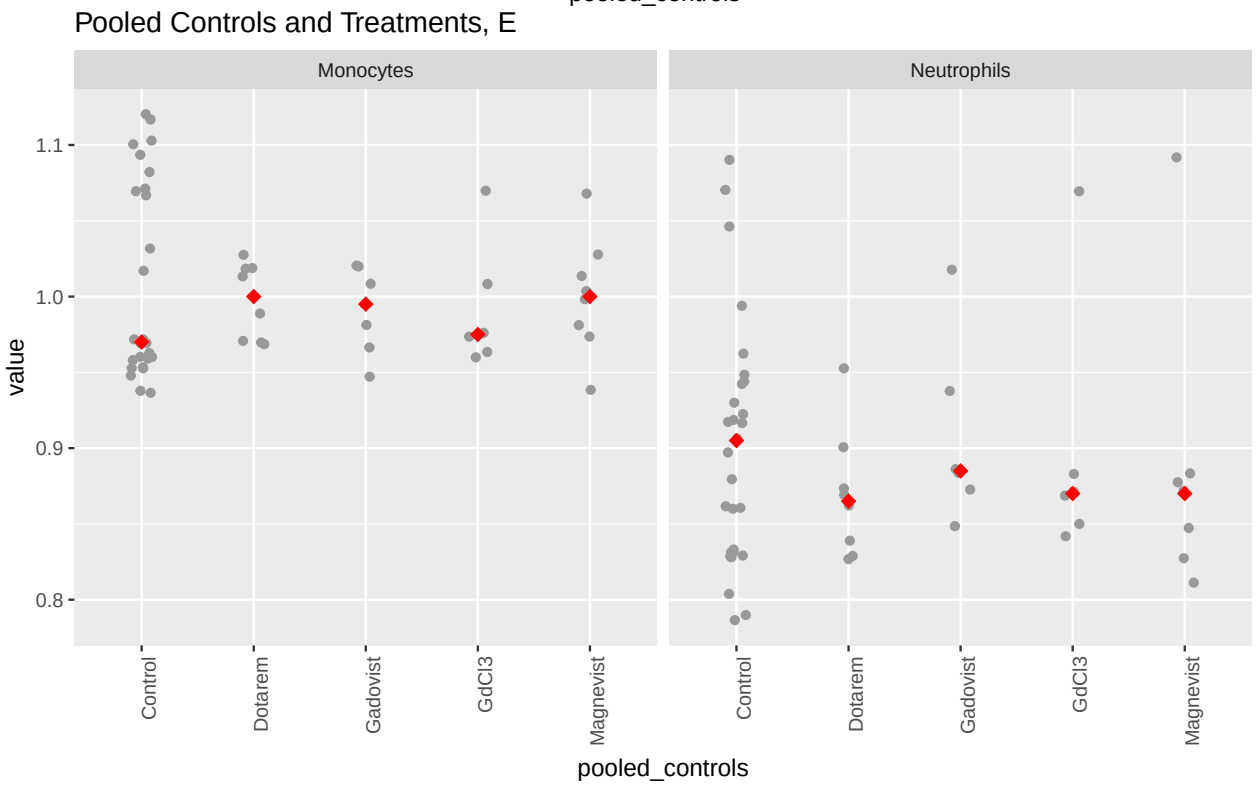
This does not seem to have changed much so data will be treated as normal for this initial study, however it appears to have some clustering.

Explore summary stats of Young's modulus and deformation, keeping each trial separate (in principle the controls can be pooled and will be below). There are many outliers so the effect is clearest using a robust statistic (median) for the central tendency (red dot):



Exploring ranges with all controls pooled:





Def and E continue to be "mirror images" of each other which is a good reality check that the process is physical. Visually the effects seem relatively clear however the distribution of the controls is highly dispersed, and some contrast agents sets contain one strong outlier.

For a first statistical analysis, to handle the large impact of outliers, the data were regressed against a linear

model with a Student's t distribution. JAGS code for the model was:

```
student_t_model <- "
model {

    for (i in 1:length(y)) {
        y[i] ~ dt(mu[i], tau, k)
        mu[i] <- beta[cont_ag[i]]
    }

    for (k in 1:n_cont_ag) {
        beta[k] ~ dnorm(mu_ag, tau_ag)
    }

    sig ~ dunif(0.001, 1000)
    tau <- 1 / sig^2
    k ~ dunif(0.001, 10)

    mu_ag ~ dnorm(0, 1 / 10000)
    sig_ag ~ dunif(0.001, 1000)
    tau_ag <- 1 / sig_ag^2

}
"
```

As both Young's modulus and deformation behave similarly we use the deformation for this first analysis. As we are not sure if the two cell types are affected similarly by the agents, we analyze Neutrophils alone first. Cleaning and subsetting the data:

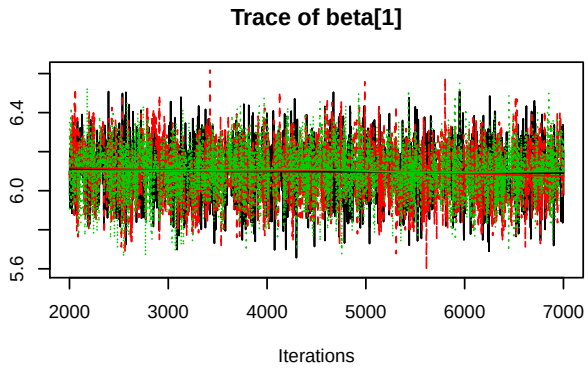
```
neutro_subset <- subset(rtdc_tall, measurement == "mean_def" & cell_type == "Neutrophils")
jags_dataset <- list(
    y = neutro_subset$value,
    cont_ag = neutro_subset$pooled_controls,
    n_cont_ag = length(levels(neutro_subset$pooled_controls))
)
jags_model <- jags.model(file = textConnection(student_t_model), data = jags_dataset, n.chains = 3)
```

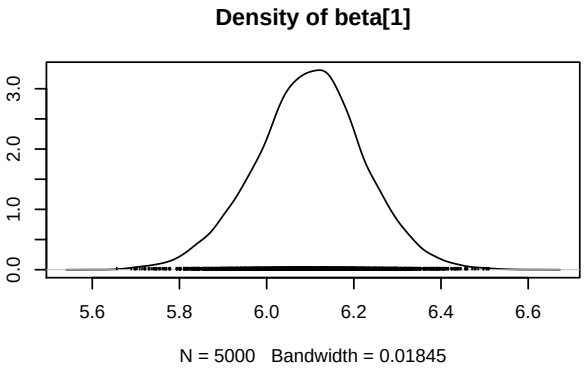
```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 56
##    Unobserved stochastic nodes: 9
##    Total graph size: 134
##
## Initializing model
```

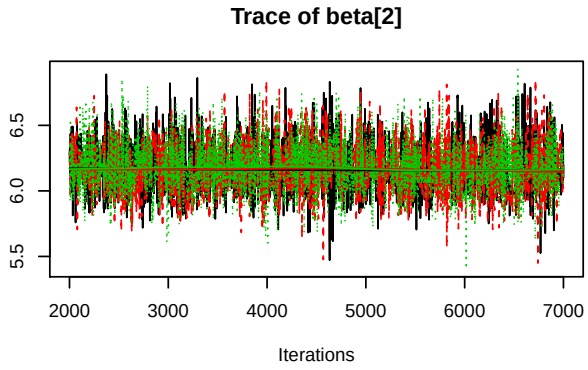
```
update(jags_model, 1e3)
sim <- coda.samples(model=jags_model, variable.names = c("beta", "tau", "k", "mu_ag", "tau_ag"), n.iter
gelman <- gelman.diag(sim)
csim <- as.mcmc(do.call(rbind, sim))
```

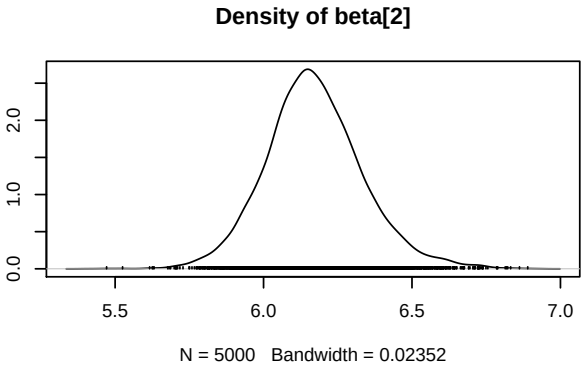
Chains look overall good, except tau_ag looks sparse:

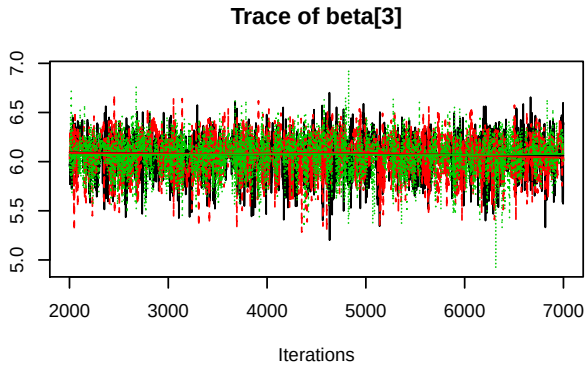
```
plot(sim)
```

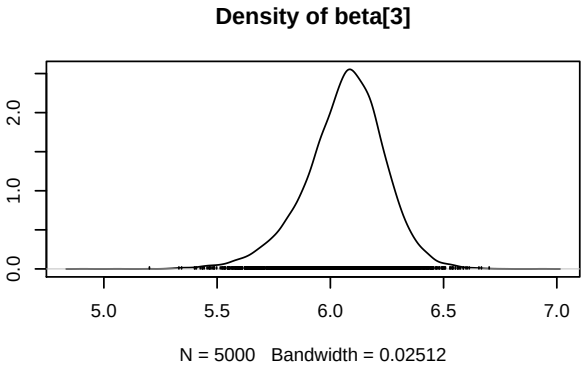

Trace of beta[1]
6.4
6.0
5.6
2000
3000
4000
5000
6000
7000
Iterations

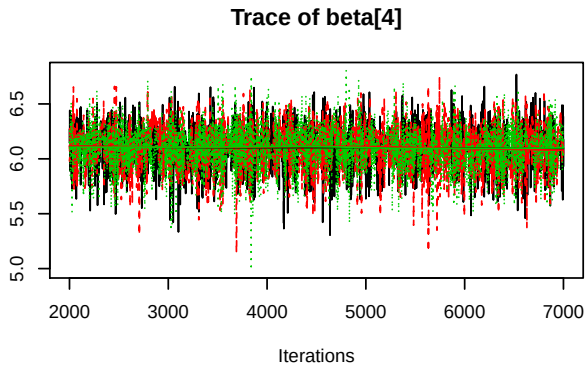

Density of beta[1]
3.0
2.0
1.0
0.0
5.6
5.8
6.0
6.2
6.4
6.6
N = 5000 Bandwidth = 0.01845

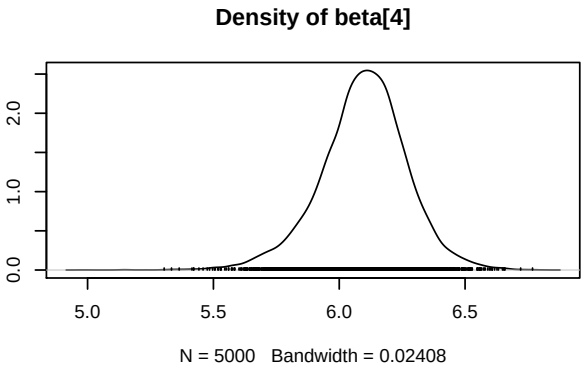

Trace of beta[2]
6.5
6.0
5.5
2000
3000
4000
5000
6000
7000
Iterations

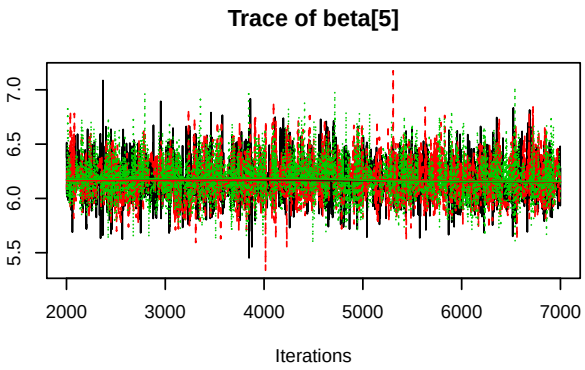

Density of beta[2]
2.0
1.0
0.0
5.5
6.0
6.5
7.0
N = 5000 Bandwidth = 0.02352

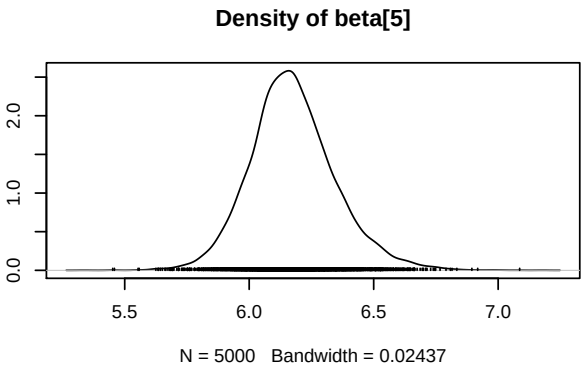

Trace of beta[3]
7.0
6.5
6.0
5.5
5.0
2000
3000
4000
5000
6000
7000
Iterations

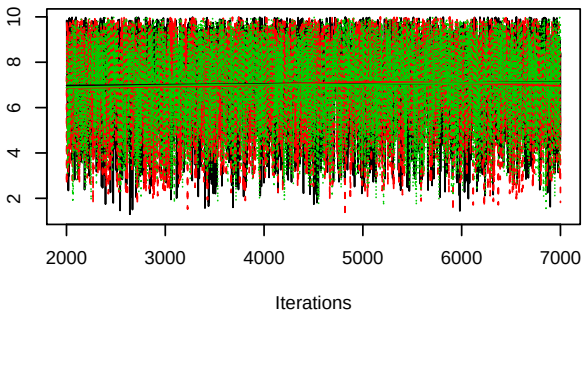

Density of beta[3]
2.0
1.0
0.0
5.0
5.5
6.0
6.5
7.0
N = 5000 Bandwidth = 0.02512

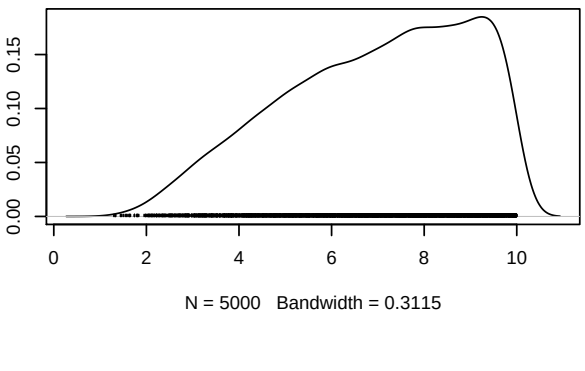

Trace of beta[4]
6.5
6.0
5.5
5.0
2000
3000
4000
5000
6000
7000
Iterations

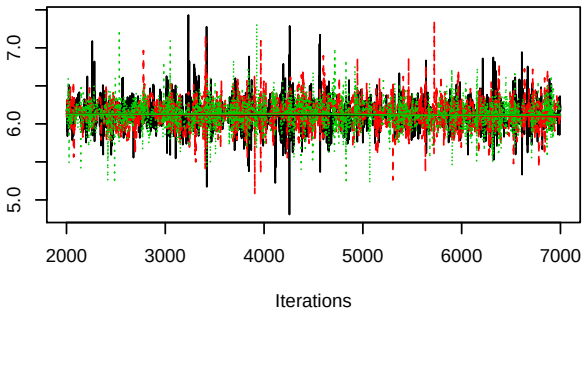

Density of beta[4]
2.0
1.0
0.0
5.0
5.5
6.0
6.5
N = 5000 Bandwidth = 0.02408

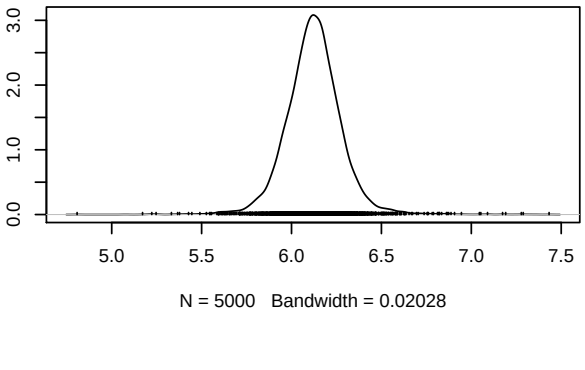

Trace of beta[5]
5.5
6.0
6.5
7.0
2000
3000
4000
5000
6000
7000
Iterations

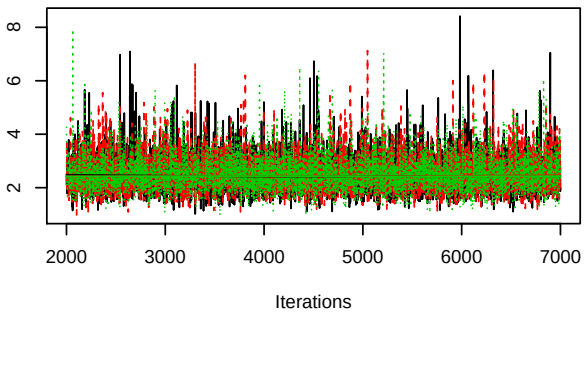

Density of beta[5]
0.0
1.0
2.0
5.5
6.0
6.5
7.0
N = 5000 Bandwidth = 0.02437

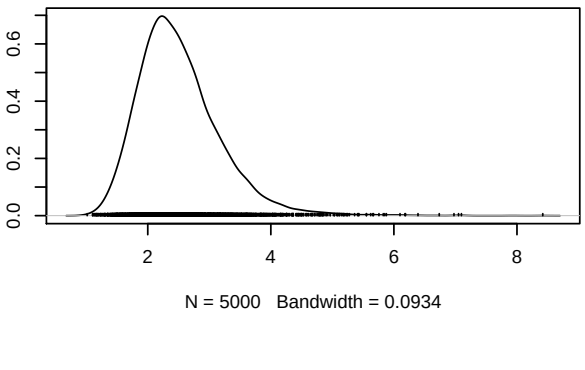

Trace of k
2
4
6
8
10
2000
3000
4000
5000
6000
7000
Iterations

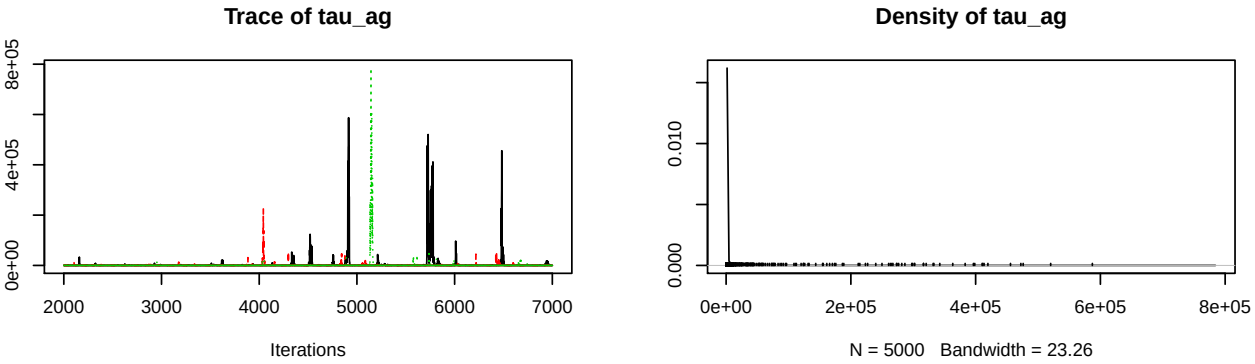

Density of k
0.00
0.05
0.10
0.15
0
2
4
6
8
10
N = 5000 Bandwidth = 0.3115


Trace of mu_ag
5.0
6.0
7.0
2000
3000
4000
5000
6000
7000
Iterations


Density of mu_ag
0.0
1.0
2.0
3.0
5.0
5.5
6.0
6.5
7.0
7.5
N = 5000 Bandwidth = 0.02028


Trace of tau
2
4
6
8
2000
3000
4000
5000
6000
7000
Iterations


Density of tau
0.0
0.2
0.4
0.6
2
4
6
8
N = 5000 Bandwidth = 0.0934

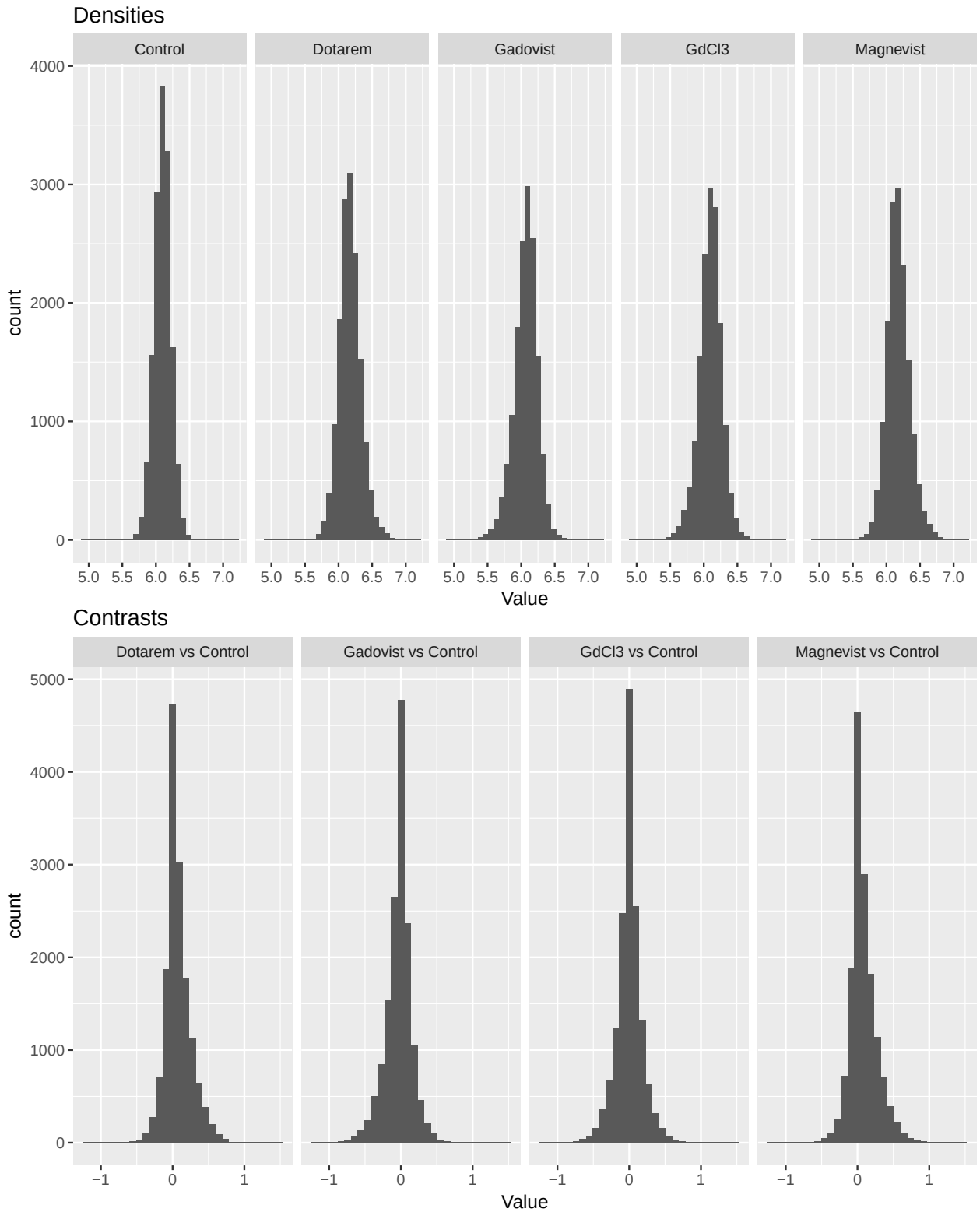


Gelman diagnostic looks good besides tau_ag:

```
print(gelman)
```

```
## Potential scale reduction factors:
##
##         Point est. Upper C.I.
## beta[1]       1.00       1.00
## beta[2]       1.00       1.00
## beta[3]       1.00       1.00
## beta[4]       1.00       1.00
## beta[5]       1.00       1.00
## k             1.00       1.00
## mu_ag         1.00       1.00
## tau           1.00       1.00
## tau_ag        1.11       1.14
##
## Multivariate psrf
##
## 1.01
```

Plot densities and contrasts:



The contrast densities seem firmly centered on zero, nevertheless we show the high-density intervals in a table

```
print("Densities:")
```

```
## [1] "Densities:"
```

```
print(hdi(densities))
```

```
##        Control  Dotarem Gadovist    GdCl3 Magnevist
## lower 5.836074 5.850738 5.681042 5.723118  5.841401
## upper 6.332603 6.509355 6.391391 6.429991  6.528930
## attr(,"credMass")
## [1] 0.95
```

```
print("Contrasts:")
```

```
## [1] "Contrasts:"
```

```
print(hdi(contrasts))
```

```
##       Dotarem vs Control Gadovist vs Control GdCl3 vs Control
## lower         -0.2483533          -0.4341439       -0.4070435
## upper          0.4871676           0.3299418        0.3611116
##       Magnevist vs Control
## lower           -0.2683354
## upper            0.4828676
## attr(,"credMass")
## [1] 0.95
```

TO DO: - It is surprising that the HDI of the control is comparable to (or less than) the HDIs of some of the contrast agents. It should be much wider, and presumably this is a shrinkage effect, but needs to be investigated further. - tau_ag is not converging, should I rethink this aspect of the model (assuming this is a model worth keeping?)