MCMC Diagnostics - IFLS data

Sarah Teichman

04/13/2020

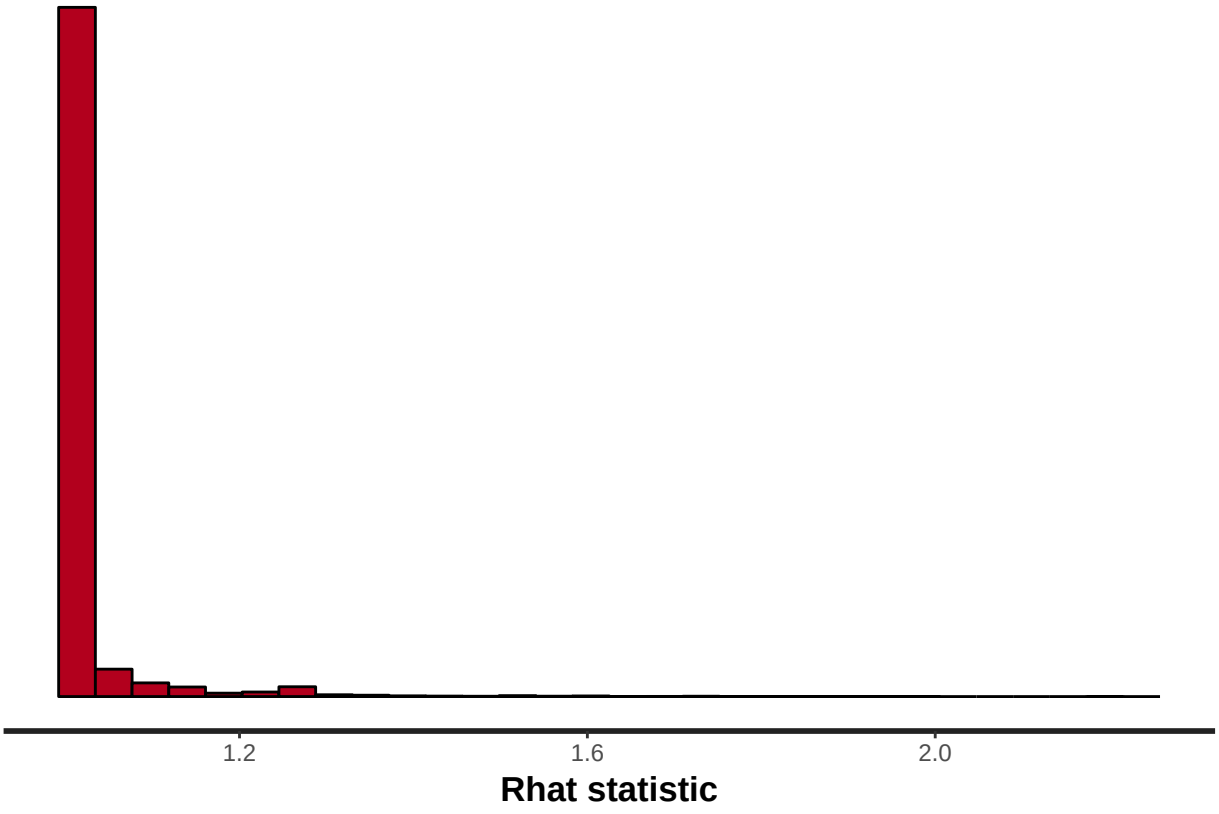
```
K <- 7
Ti <- 3
N <- 1973
```

General MCMC diagnostic plots

Overall model diagnostics from rstan package.

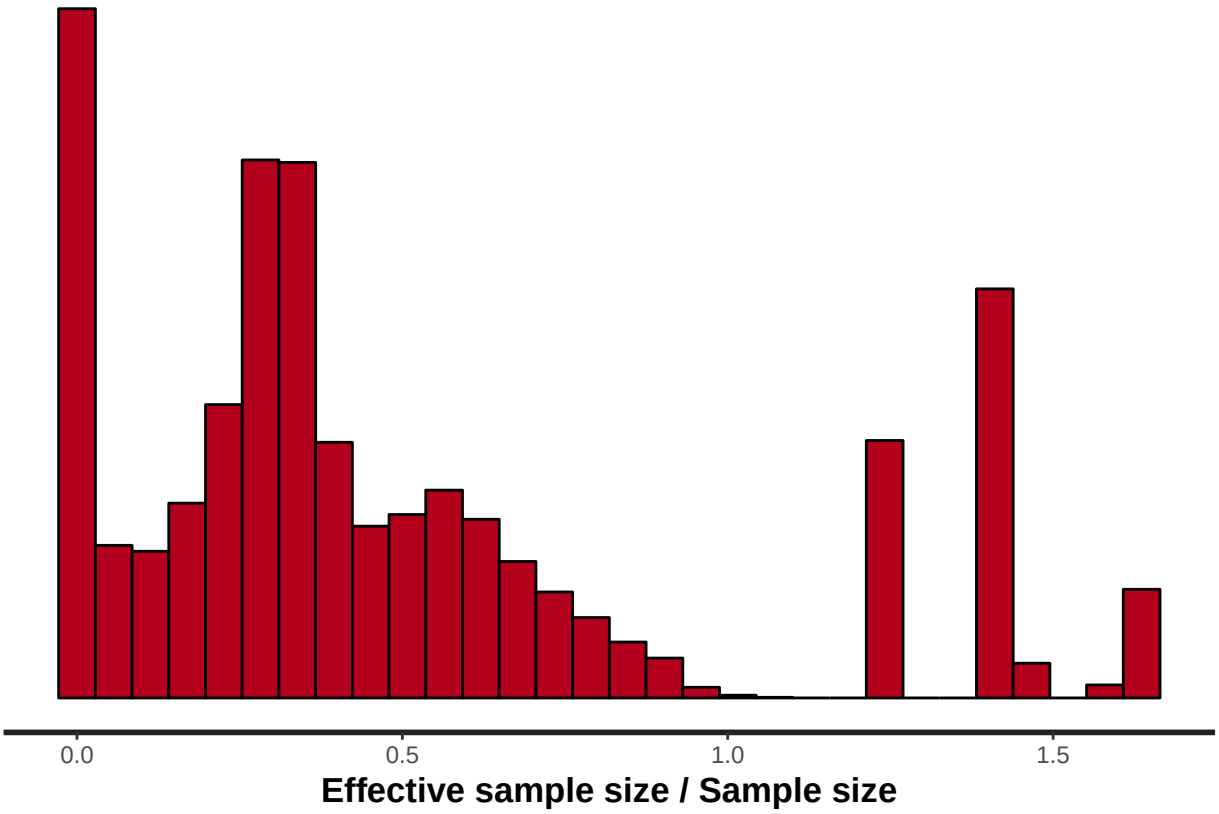
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 1 rows containing non-finite values (stat_bin).
```




```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 2 rows containing non-finite values (stat_bin).
```




```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 4 rows containing non-finite values (stat_bin).
```

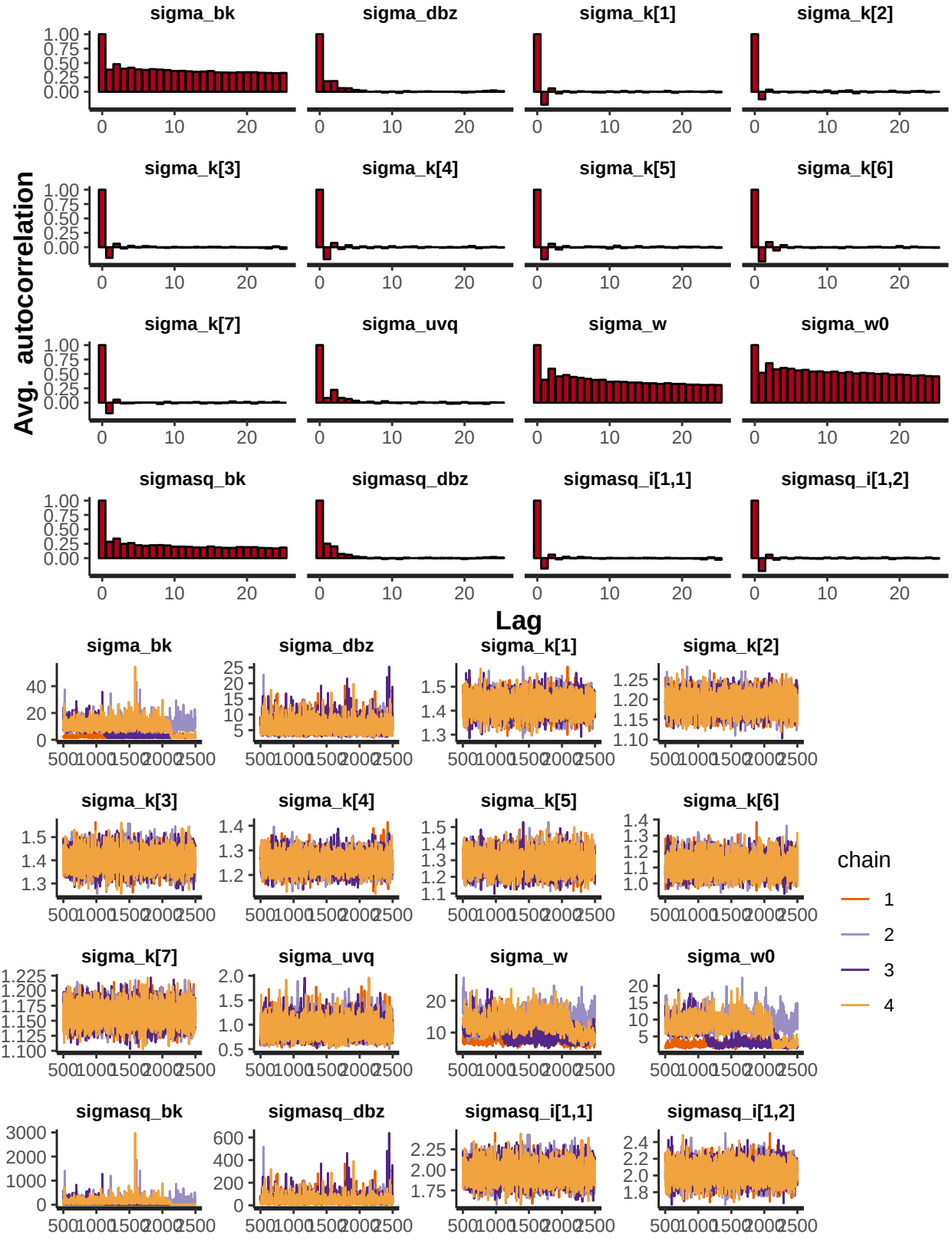
Individual Parameter Diagnostics

Individual parameter plots. Autocorrelation and trace plots for individual parameters, and histograms of posterior medians for group parameters.

```
get_single_plots <- function(fit, param) {
  print(fit_summ[param,c(1,2,3,5,6,7,9,10)])
  print(stan_ac(fit, pars = param))
  print(rstan::traceplot(fit, pars = param))
}
get_aggreg_plots <- function(fit, param, trim = F, trim_amount) {
  ind <- grep(paste0("^",param), rownames(as.data.frame(summary(fit)$summary)))
  medians <- data.frame(avg = as.data.frame(summary(fit)$summary)$`50%`[ind])
  print(paste0("Summary statistics for posterior medians of ",param))
  print(summary(medians))
  title <- paste0("Posterior Medians of ",param)
  print(ggplot(medians, aes(x = avg)) + geom_histogram(bins = 60) + ggtitle(title))
  if (trim == T) {
    lim <- quantile(abs(medians$avg), probs = trim_amount)
    meds_trim <- medians %>% filter(abs(medians$avg) < lim)
    print(ggplot(meds_trim, aes(x = avg)) + geom_histogram(bins = 60) +
            ggtitle(paste0(title, " Without Extreme ",100*(1-trim_amount),"%")))
  }
}
plot_fit <- function(fit) {
  get_single_plots(fit, sigma_params)
  get_single_plots(fit, beta_k)
```

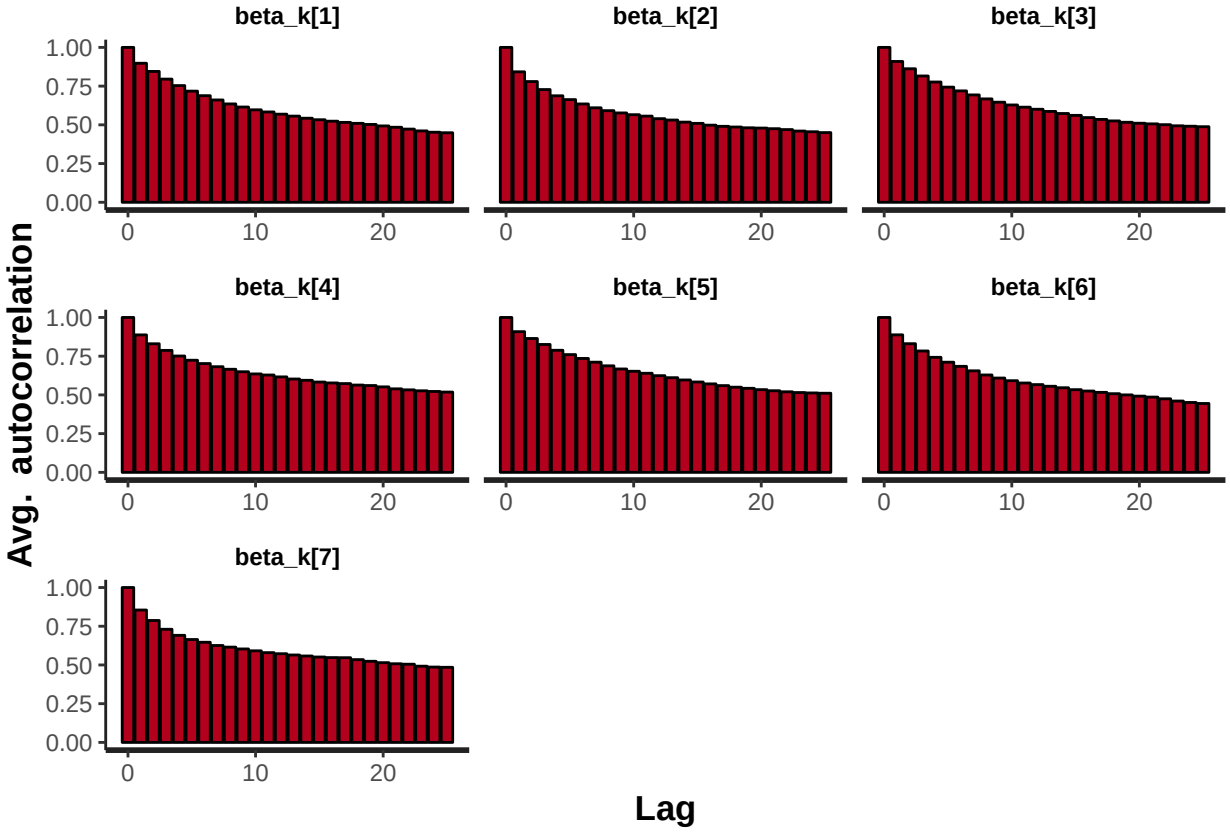
```
  get_single_plots(fit, other_1d)
  get_single_plots(fit, u)
  get_single_plots(fit, v)
  get_single_plots(fit, q)
  get_aggreg_plots(fit, "w")
  get_aggreg_plots(fit, "z")
  get_aggreg_plots(fit, "p")
  get_aggreg_plots(fit, "eta", trim = T, trim_amount = .60)
  get_aggreg_plots(fit, "lambda", trim = T, trim_amount = .60)
  get_aggreg_plots(fit, "kappa", trim = T, trim_amount = .60)
}
plot_fit(fit)
```

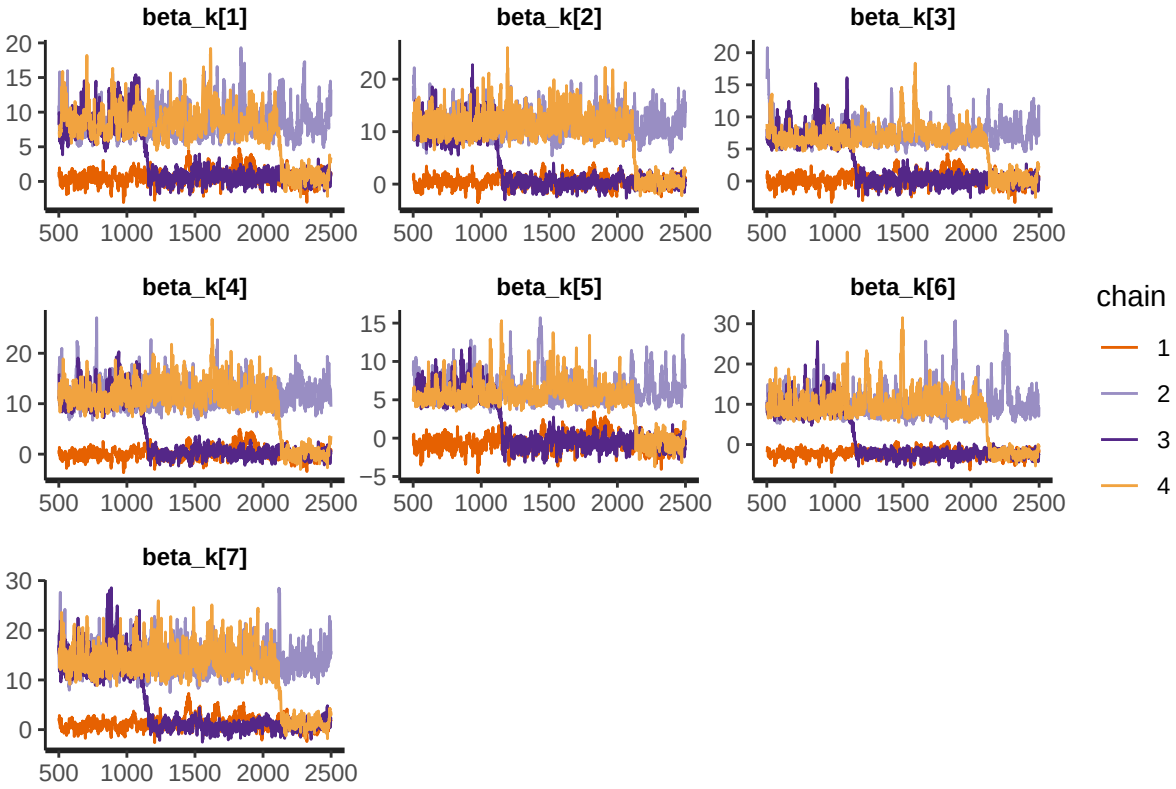
```
##                      mean      se_mean           sd       25%        50%
## sigma_bk        7.0048720 2.371589e+00   5.04895919  2.363871  6.7661646
## sigma_dbz       6.1396969 3.525321e-02   2.13695049  4.677695  5.6685910
## sigma_k[1]      1.4225211 3.393753e-04   0.03883003  1.396277  1.4221105
## sigma_k[2]      1.1892623 2.307172e-04   0.02323651  1.173091  1.1892327
## sigma_k[3]      1.4000941 4.232555e-04   0.04208664  1.371450  1.3988075
## sigma_k[4]      1.2504725 3.272360e-04   0.03496144  1.226482  1.2495956
## sigma_k[5]      1.2857090 5.335143e-04   0.05803654  1.245965  1.2837277
## sigma_k[6]      1.1135869 4.977397e-04   0.05660395  1.074718  1.1106749
## sigma_k[7]      1.1606669 1.601136e-04   0.01687873  1.149158  1.1603570
## sigma_uvq       0.9044773 2.867962e-03   0.18000791  0.776693  0.8802021
## sigma_w        11.1717180 1.451997e+00   3.34001348  8.405610 10.7435820
## sigma_w0        5.9108454 1.625466e+00   3.42406531  2.666250  5.8542585
## sigmasq_bk     74.5570342 3.586037e+01 104.78672445  5.587887 45.7809885
## sigmasq_dbz    42.2618640 6.143796e-01  35.43566266 21.880830 32.1329244
## sigmasq_i[1,1]  1.9620347 1.194632e-03   0.11810515  1.880876  1.9566625
## sigmasq_i[1,2]  2.0250738 9.684559e-04   0.11064572  1.949589  2.0223982
##                      75%        n_eff      Rhat
## sigma_bk        10.568620     4.532364 1.5253770
## sigma_dbz        7.125118  3674.443255 1.0016586
## sigma_k[1]       1.447667 13091.064940 0.9998406
## sigma_k[2]       1.204702 10143.360730 0.9995889
## sigma_k[3]       1.428114  9887.426478 0.9998228
## sigma_k[4]       1.273467 11414.487738 0.9998574
## sigma_k[5]       1.323858 11833.435055 0.9997391
## sigma_k[6]       1.149802 12932.688535 0.9998858
## sigma_k[7]       1.171956 11112.794209 0.9998475
## sigma_uvq        1.003097  3939.456448 0.9997158
## sigma_w         13.434599     5.291331 1.3864877
## sigma_w0         8.456378     4.437398 1.5663199
## sigmasq_bk     111.695725     8.538527 1.1979273
## sigmasq_dbz     50.767307  3326.653021 1.0018716
## sigmasq_i[1,1]   2.039509  9773.937604 0.9998137
## sigmasq_i[1,2]   2.095740 13052.975178 0.9998493
```



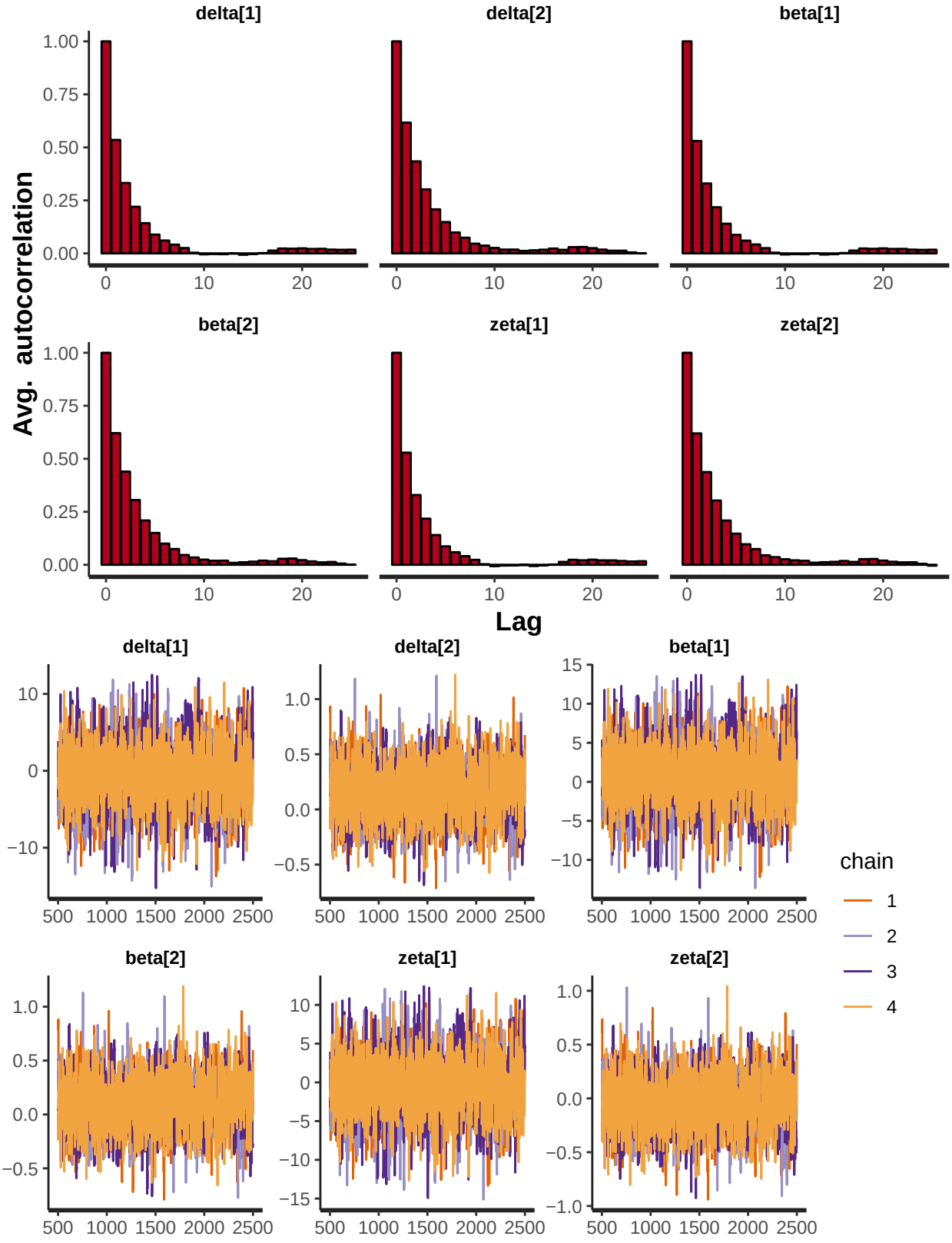

```
##                  mean   se_mean       sd        25%      50%       75%
## beta_k[1] 4.900939 2.151239 4.305230  0.7415360  5.862923  8.292546
## beta_k[2] 6.256661 3.030949 5.711108  0.5855202  8.152291 11.173689
```

```
## beta_k[3] 3.993682 1.911224 3.766290  0.3149156  5.400950  6.980279
## beta_k[4] 6.458211 3.209188 6.146983  0.2814340  8.608669 11.722747
## beta_k[5] 3.039957 1.842125 3.591854 -0.4377192  4.360686  5.799288
## beta_k[6] 4.255333 3.371043 6.574448 -2.0741297  6.586086  9.248415
## beta_k[7] 8.075100 3.579523 6.857376  1.1519448 10.413559 13.979482
##              n_eff     Rhat
## beta_k[1] 4.005120 1.744096
## beta_k[2] 3.550450 1.873766
## beta_k[3] 3.883332 1.776610
## beta_k[4] 3.668882 1.856235
## beta_k[5] 3.801887 1.764381
## beta_k[6] 3.803559 1.796344
## beta_k[7] 3.669996 1.894378
```



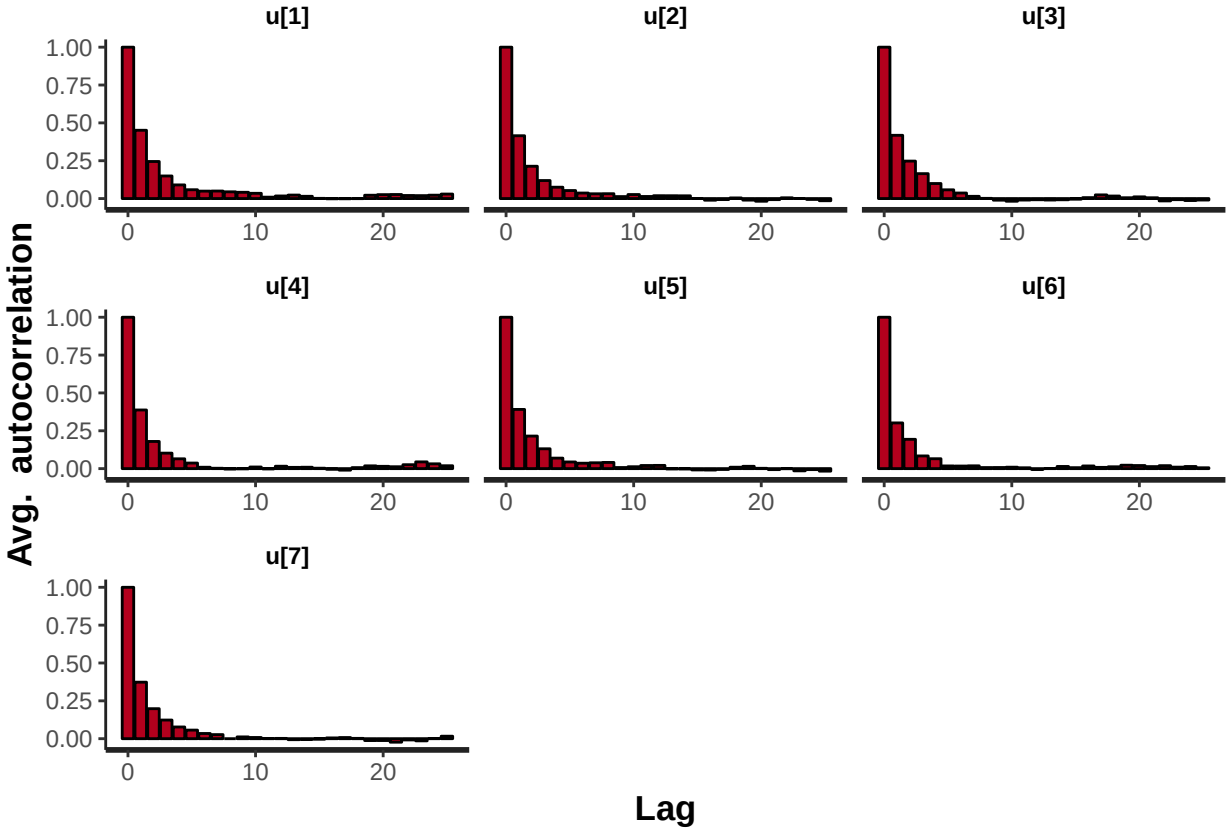


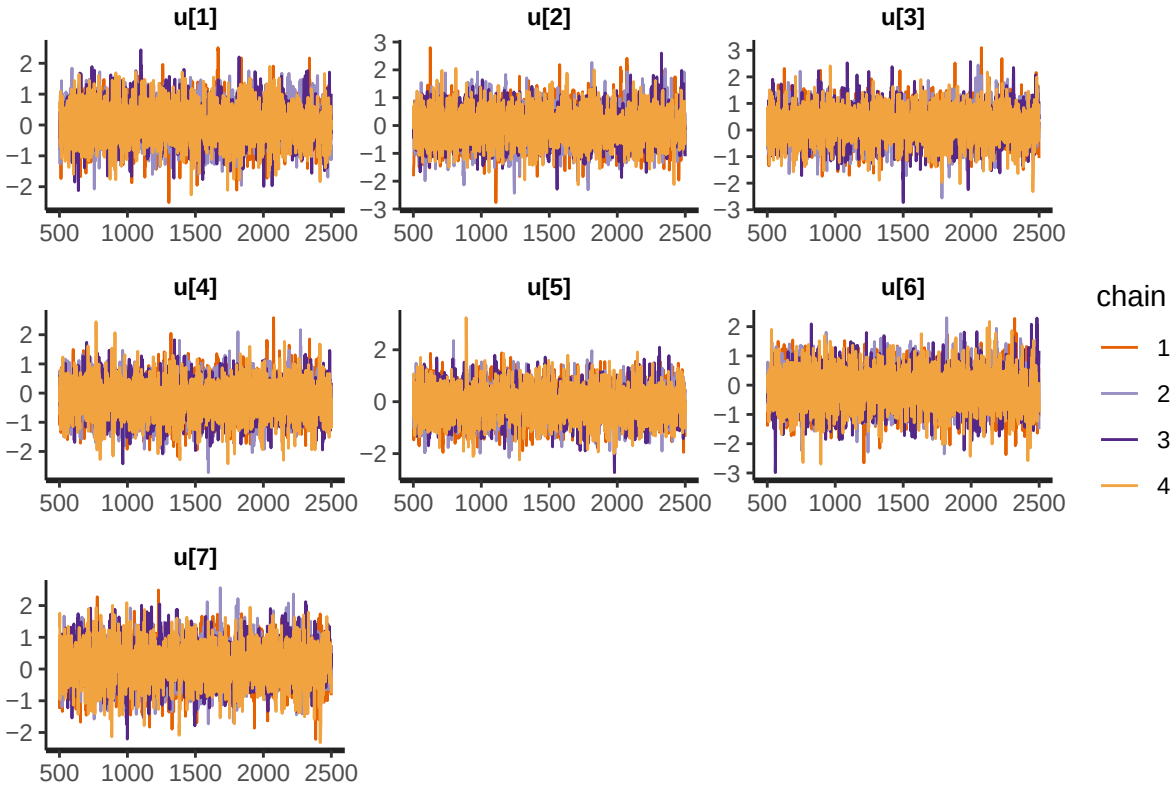

```
##                  mean     se_mean        sd          25%         50%
## delta[1] -0.54309041 0.084539377 3.7329599 -2.796875227 -0.52238940
## delta[2]  0.15888741 0.006213737 0.2347999  0.003089784  0.15914380
## beta[1]   0.95416746 0.084180062 3.7320764 -1.289576357  0.96853271
## beta[2]   0.09470489 0.006186447 0.2338393 -0.061989847  0.09302878
## zeta[1]  -0.44490865 0.083954741 3.7316494 -2.691260579 -0.44046160
## zeta[2]  -0.05250057 0.006200032 0.2352066 -0.207057043 -0.05401308
##                 75%    n_eff     Rhat
## delta[1] 1.7493138 1949.793 1.002034
## delta[2] 0.3152038 1427.874 1.004112
## beta[1]  3.2478754 1965.543 1.002004
## beta[2]  0.2497078 1428.738 1.004014
## zeta[1]  1.8414703 1975.656 1.001992
## zeta[2]  0.1046369 1439.167 1.004021
```



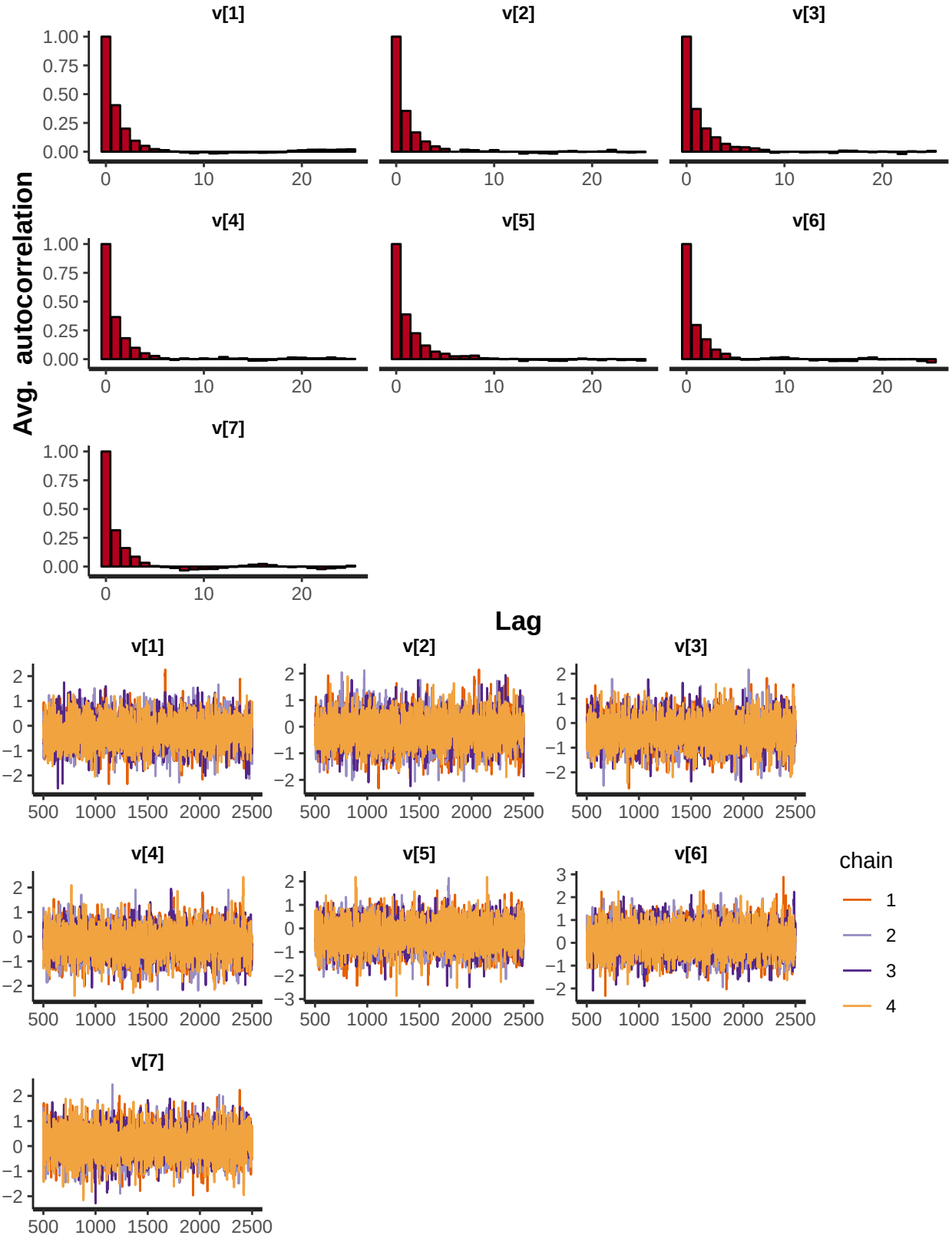

```
##              mean     se_mean        sd        25%          50%       75%
## u[1]  0.05987843 0.01294488 0.6082523 -0.3445463  0.055832693 0.4537761
## u[2] -0.01034611 0.01234163 0.6149225 -0.4169832 -0.007352468 0.3938160
```

```
## u[3]  0.19230955 0.01200907 0.6122103 -0.2055332  0.197619917 0.5845383
## u[4] -0.19950755 0.01092436 0.6083229 -0.6012162 -0.200828573 0.2038311
## u[5] -0.05784834 0.01192924 0.6078735 -0.4504016 -0.055330681 0.3443882
## u[6] -0.13605457 0.01113249 0.6335521 -0.5491582 -0.134401597 0.2691507
## u[7]  0.17223626 0.01155138 0.6040749 -0.2138975  0.174355694 0.5663861
##         n_eff     Rhat
## u[1] 2207.860 1.001394
## u[2] 2482.537 1.000807
## u[3] 2598.859 1.000922
## u[4] 3100.819 1.001651
## u[5] 2596.576 1.002121
## u[6] 3238.770 1.000360
## u[7] 2734.728 1.002374
```



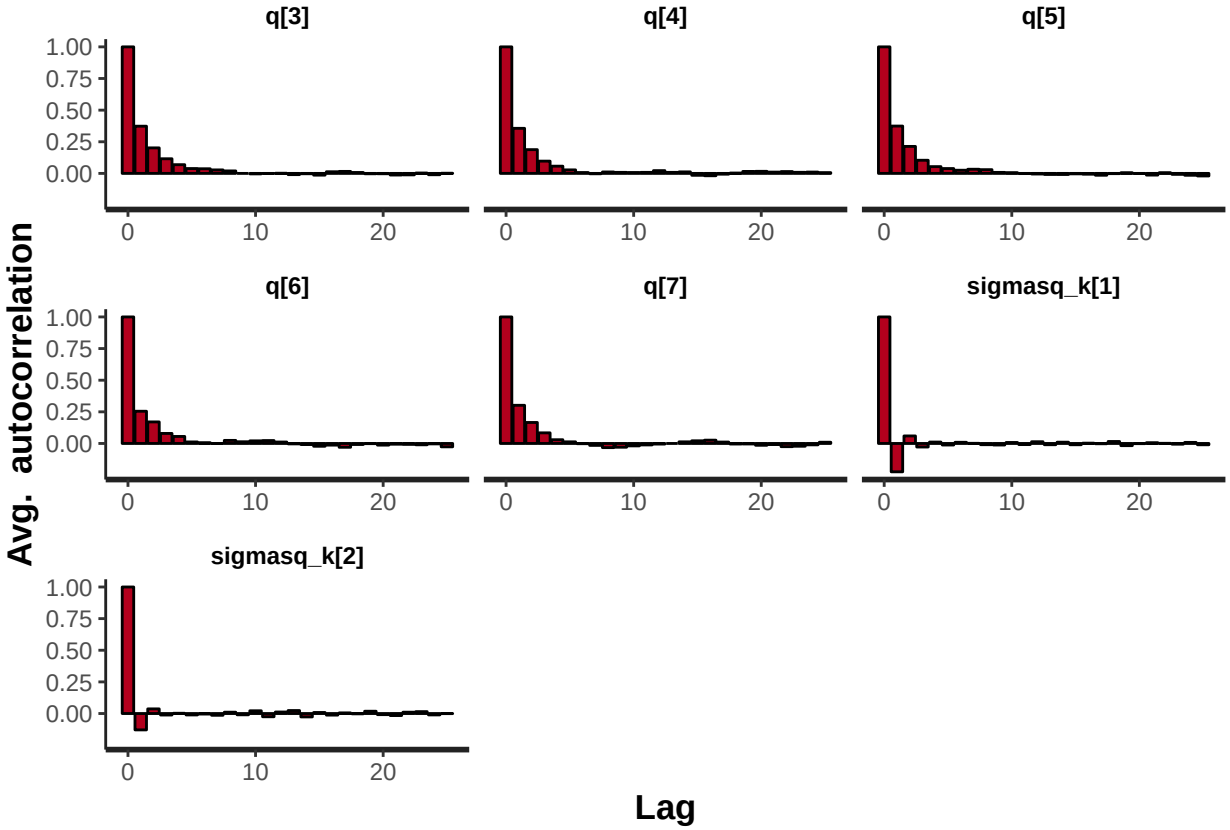


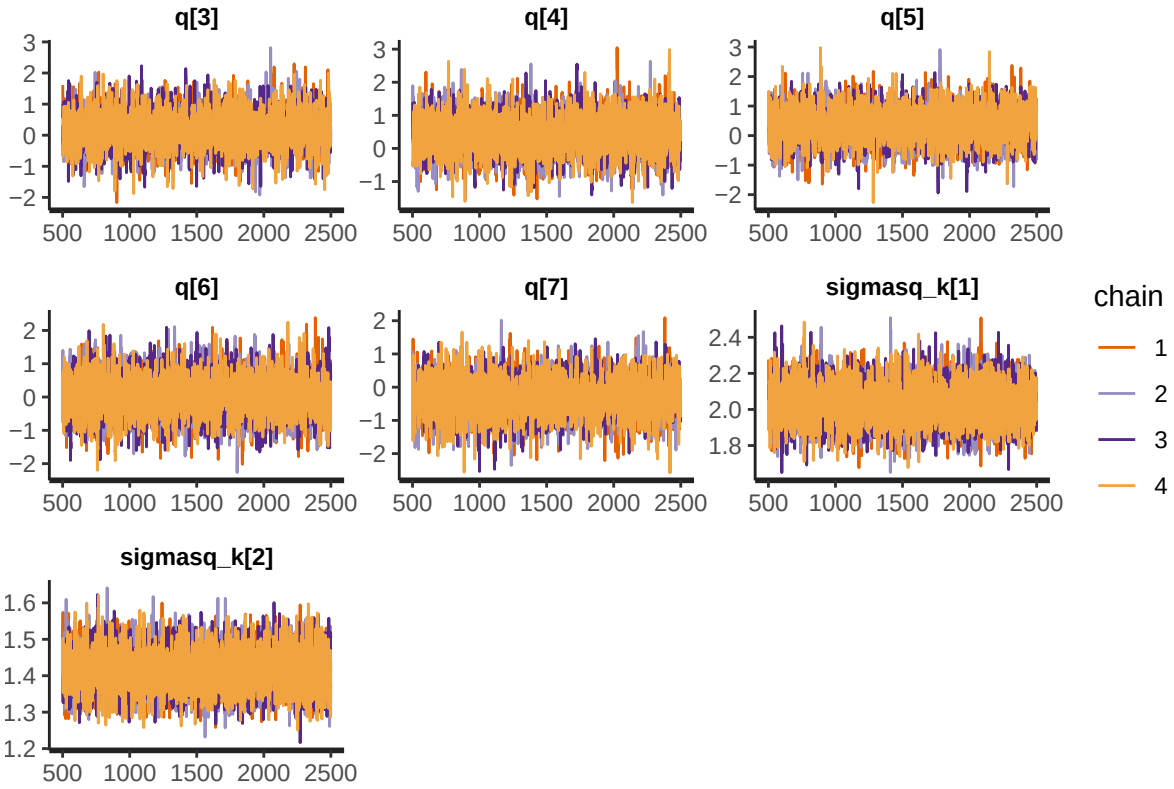

```
##             mean     se_mean        sd        25%        50%        75%
## v[1] -0.2550317 0.010027170 0.5537652 -0.6262088 -0.2541962 0.10068801
## v[2] -0.1418473 0.009704091 0.5538544 -0.5039297 -0.1464898 0.22086703
## v[3] -0.3513558 0.010411944 0.5595710 -0.7107788 -0.3506599 0.01020299
## v[4] -0.2482662 0.009942023 0.5676659 -0.6214295 -0.2493175 0.12994134
## v[5] -0.3027896 0.010809881 0.5658382 -0.6726487 -0.3027278 0.06133363
## v[6]  0.1193779 0.009817676 0.5870551 -0.2658443  0.1055858 0.50316842
## v[7]  0.1407047 0.009185744 0.5544574 -0.2107578  0.1474538 0.50011790
##         n_eff     Rhat
## v[1] 3049.963 0.999882
## v[2] 3257.477 1.000323
## v[3] 2888.329 1.000775
## v[4] 3260.139 1.000348
## v[5] 2739.950 1.001258
## v[6] 3575.530 1.000557
## v[7] 3643.407 1.002448
```



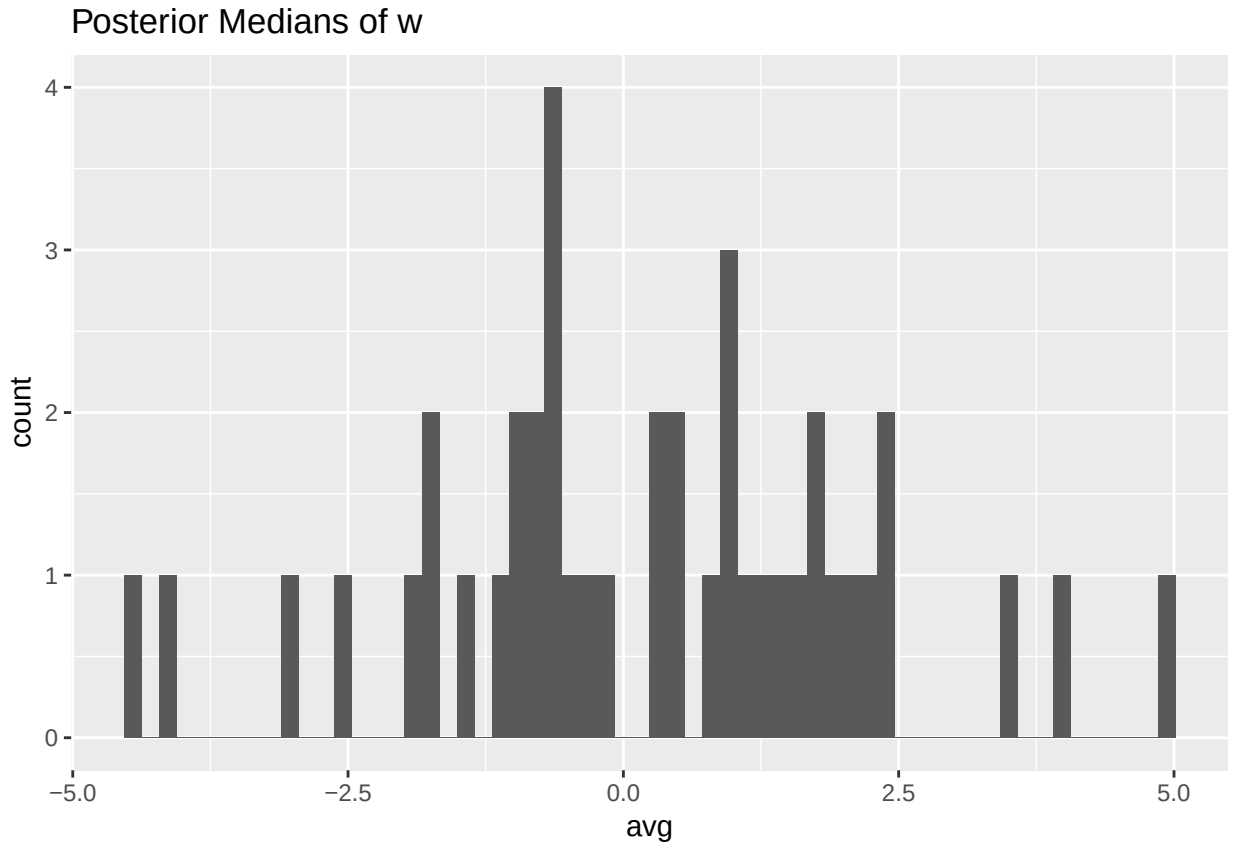

```
##                  mean       se_mean          sd         25%         50%
## q[3]       0.17435935 0.0104561157 0.56215223 -0.19123966  0.17349571
## q[4]       0.43325204 0.0100097353 0.57056449  0.06086900  0.42880934
```

```
## q[5]          0.34577457 0.0107008904 0.57102720 -0.02134057  0.33914844
## q[6]          0.02685084 0.0099070188 0.59693489 -0.37074369  0.02319073
## q[7]         -0.29481171 0.0091728025 0.55498087 -0.64919508 -0.28806806
## sigmasq_k[1]  2.02507382 0.0009684559 0.11064572  1.94958861  2.02239821
## sigmasq_k[2]  1.41488472 0.0005498192 0.05532141  1.37614246  1.41427451
##                     75%     n_eff      Rhat
## q[3]          0.53556699  2890.461 1.0011421
## q[4]          0.81091354  3249.109 1.0002154
## q[5]          0.71512220  2847.566 1.0011083
## q[6]          0.40789358  3630.513 1.0006859
## q[7]          0.05911215  3660.597 1.0024701
## sigmasq_k[1] 2.09573954 13052.975 0.9998493
## sigmasq_k[2] 1.45130779 10123.873 0.9995871
```

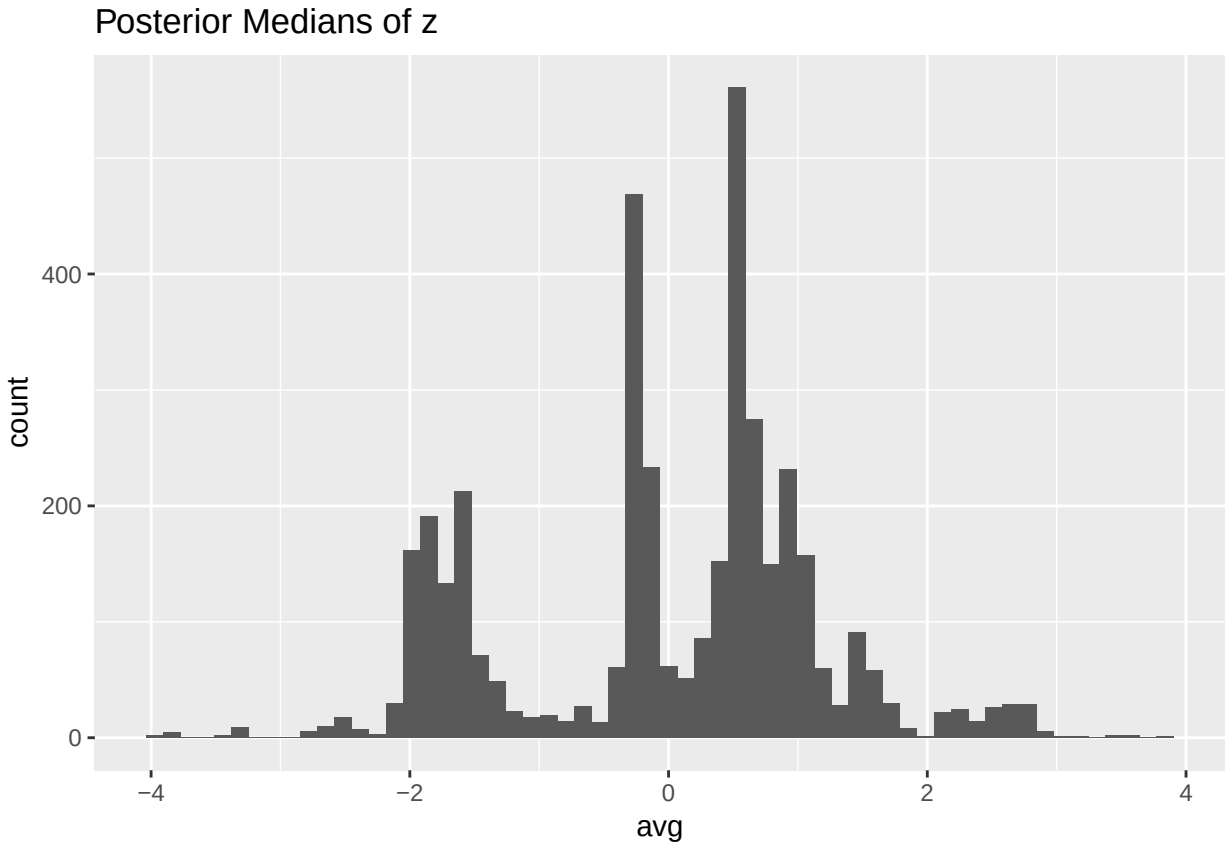




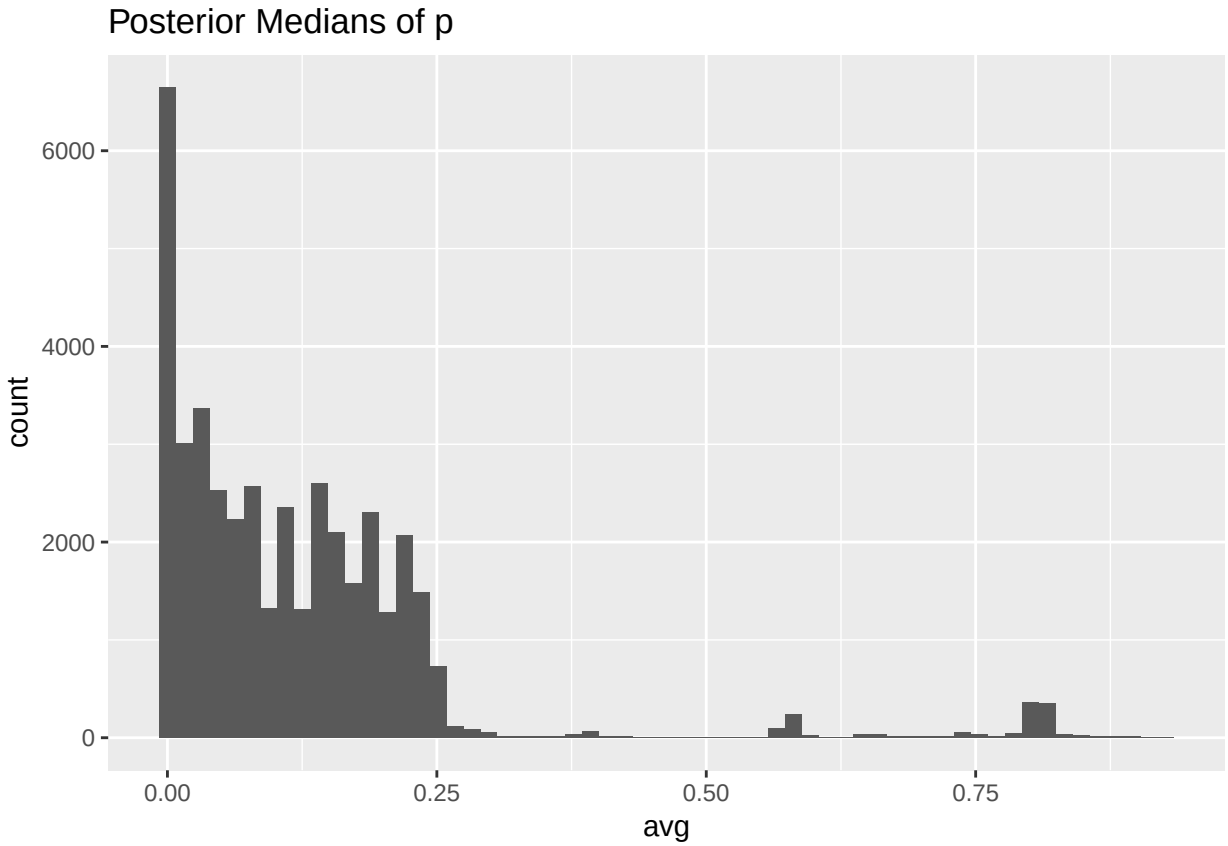

```
## [1] "Summary statistics for posterior medians of w"
##       avg
##  Min.   :-4.5017
##  1st Qu.:-0.8781
##  Median : 0.3488
##  Mean   : 0.2102
##  3rd Qu.: 1.6056
##  Max.   : 4.8899
```



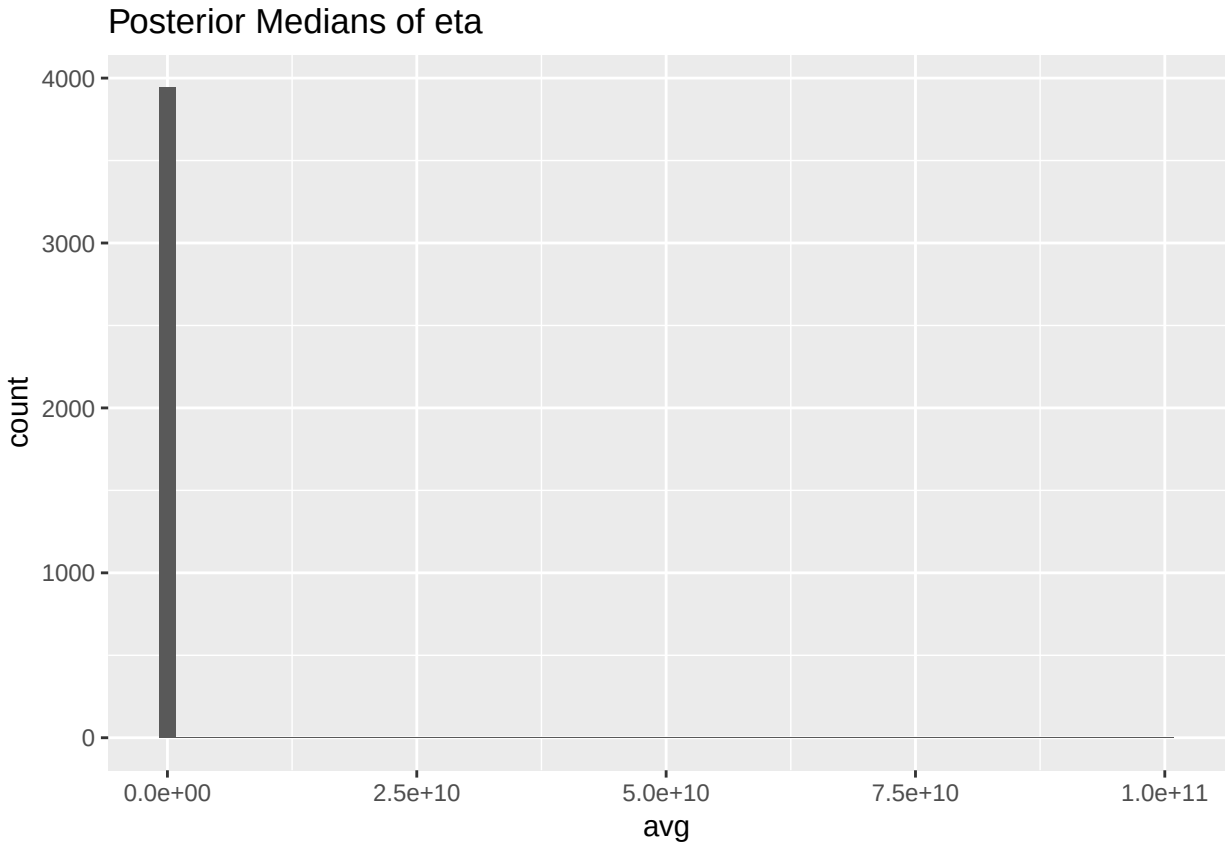

```
## [1] "Summary statistics for posterior medians of z"
##       avg
##  Min.   :-3.91899
##  1st Qu.:-0.72335
##  Median : 0.32177
##  Mean   :-0.01523
##  3rd Qu.: 0.71256
##  Max.   : 3.89118
```

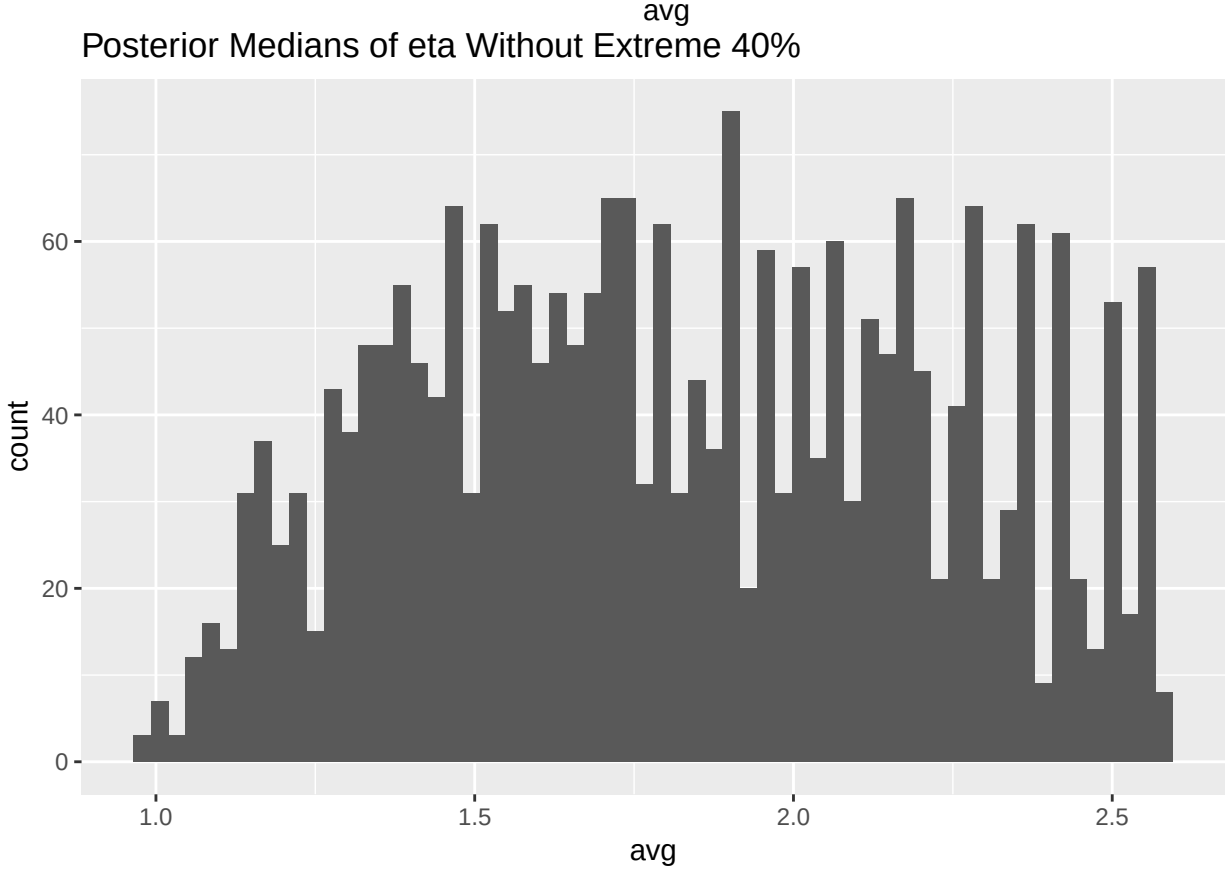



```
## [1] "Summary statistics for posterior medians of p"
##       avg
##  Min.   :0.0000005
##  1st Qu.:0.0276192
##  Median :0.0888769
##  Mean   :0.1220489
##  3rd Qu.:0.1750640
##  Max.   :0.9265146
```

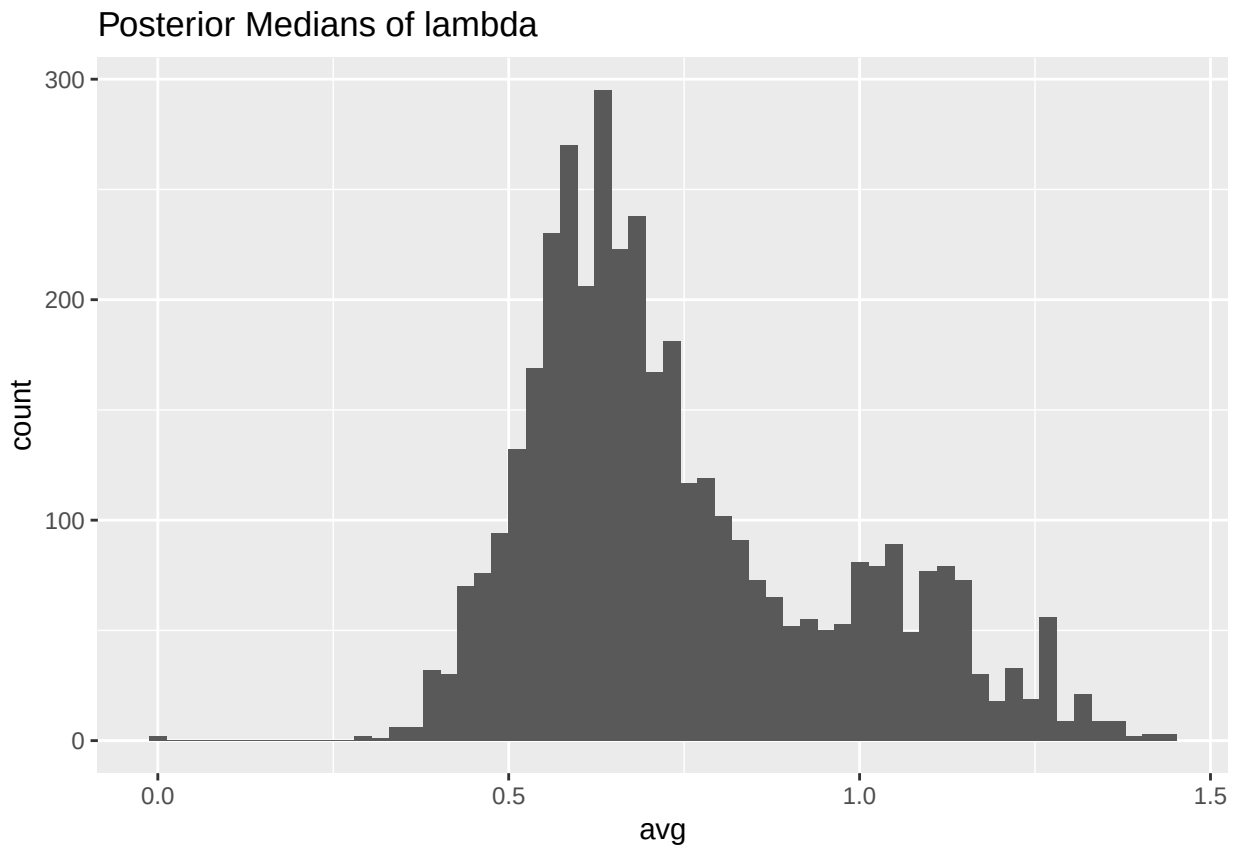


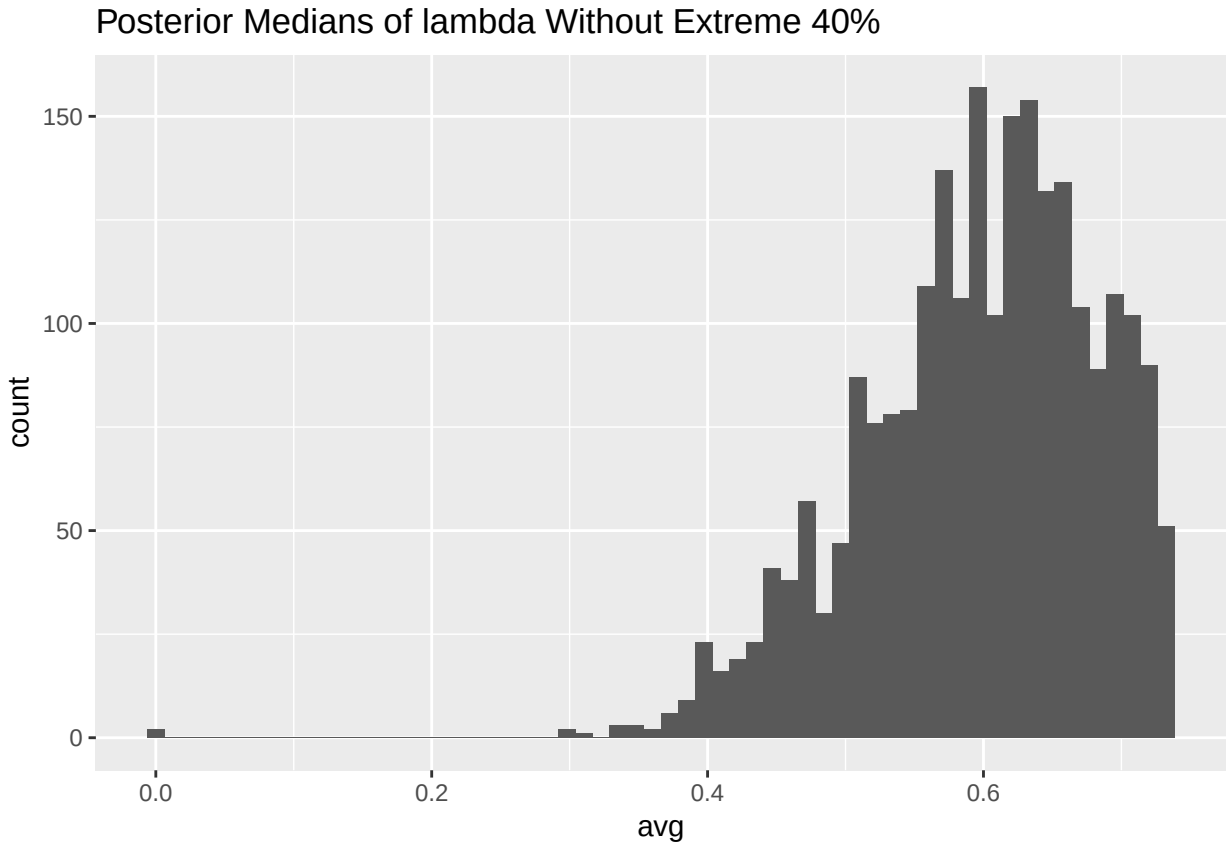

```
## [1] "Summary statistics for posterior medians of eta"
##       avg
##  Min.   :1.000e+00
##  1st Qu.:2.000e+00
##  Median :2.000e+00
##  Mean   :3.906e+07
##  3rd Qu.:3.000e+00
##  Max.   :1.001e+11
```


Posterior Medians of eta
4000
3000
2000
1000
0
count
0.0e+00
2.5e+10
5.0e+10
7.5e+10
1.0e+11
avg

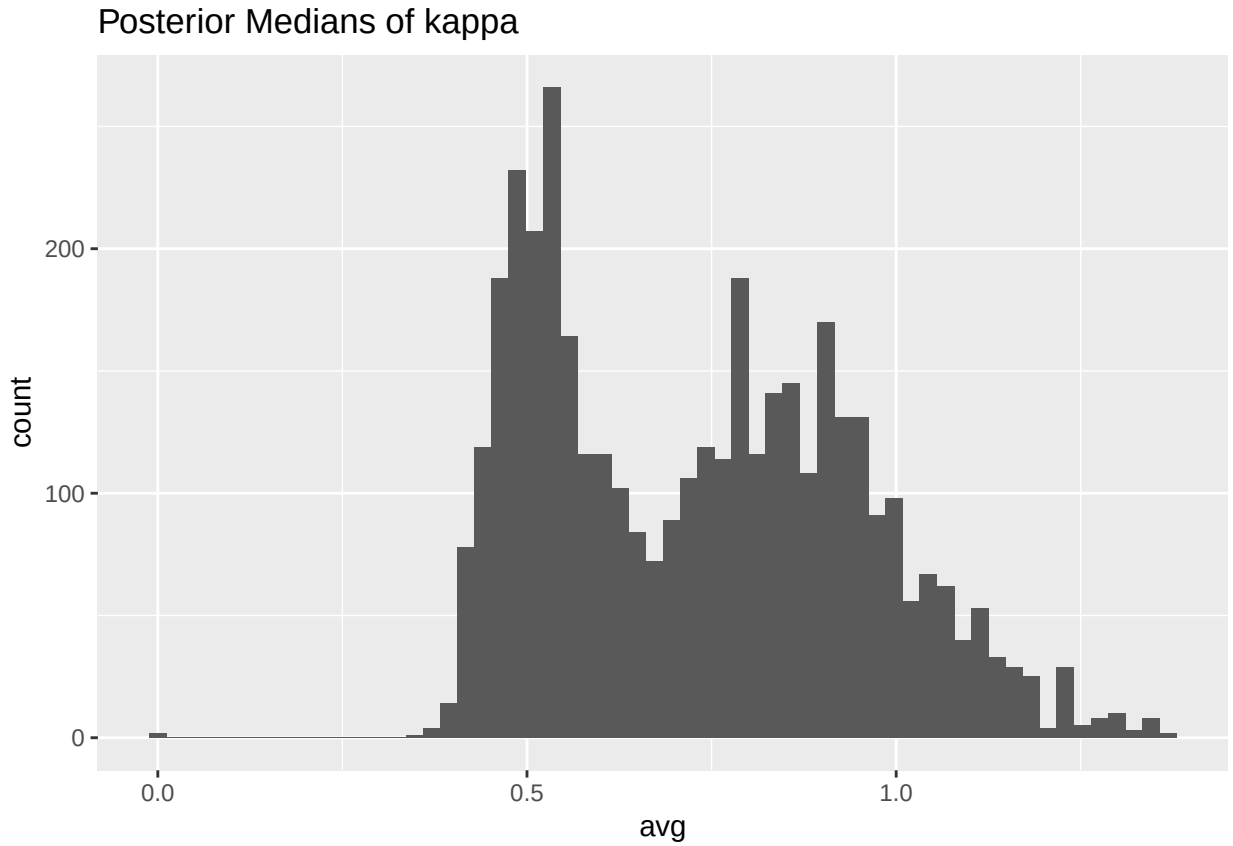

Posterior Medians of eta Without Extreme 40%
60
40
20
0
count
1.0
1.5
2.0
2.5
avg

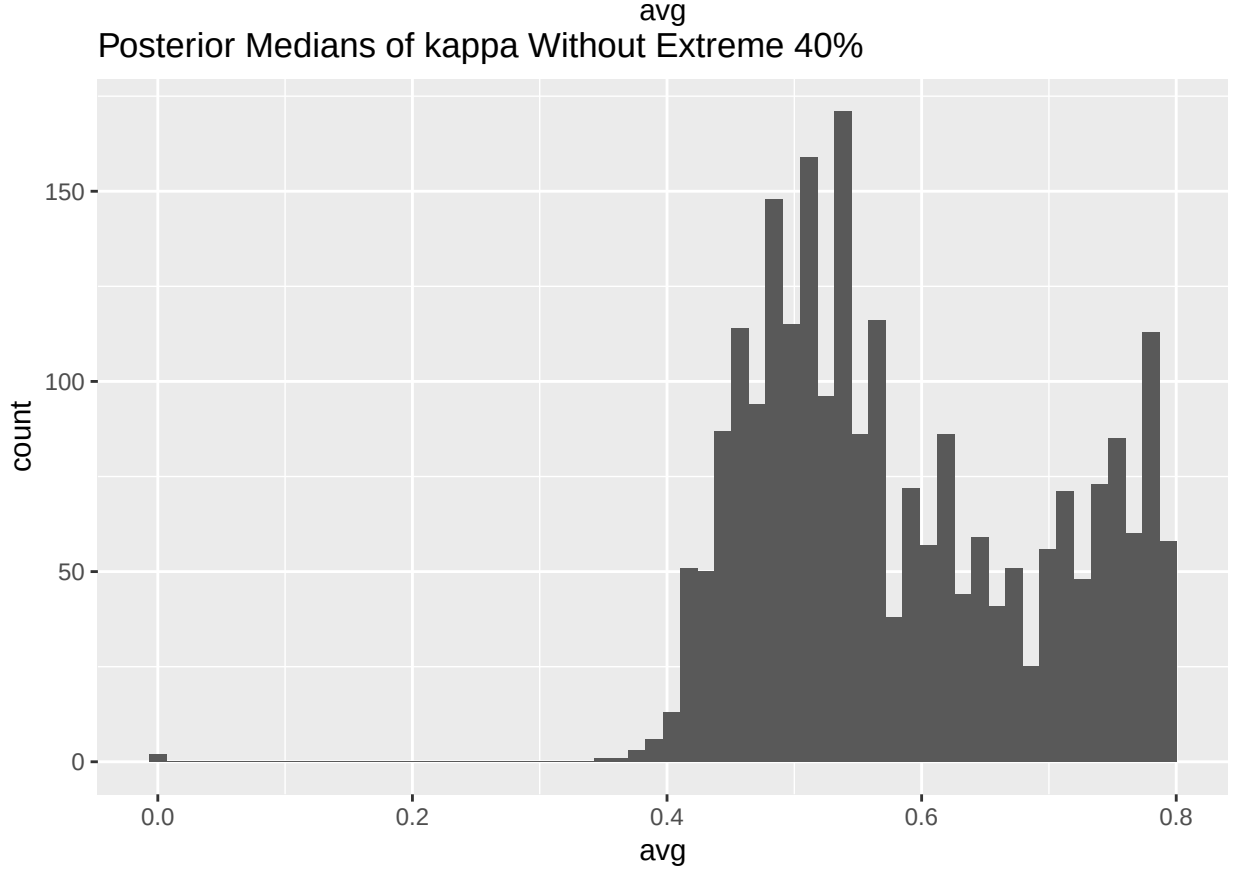
```
## [1] "Summary statistics for posterior medians of lambda"
##       avg
##  Min.   :0.0000015
##  1st Qu.:0.5890009
##  Median :0.6821466
##  Mean   :0.7468287
##  3rd Qu.:0.8777832
##  Max.   :1.4397131
```






```
## [1] "Summary statistics for posterior medians of kappa"
##       avg
##  Min.   :0.0000131
##  1st Qu.:0.5328829
##  Median :0.7347600
##  Mean   :0.7348339
##  3rd Qu.:0.9038771
##  Max.   :1.3683846
```


Posterior Medians of kappa
count
200
100
0
0.0
0.5
1.0
avg


Posterior Medians of kappa Without Extreme 40%
count
150
100
50
0
0.0
0.2
0.4
0.6
0.8
avg

Identifying Parameters with Large Rhats

```
summary(fit_summ$Rhat)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##  0.9995  1.0006  1.0022  1.0188  1.0063  2.2237       1
```

```
big_Rhat <- fit_summ$Rhat > 5
big_Rhat_dat <- fit_summ[big_Rhat,c(1,2,10)]
big_Rhat_dat
```

```
##    mean se_mean Rhat
## NA   NA      NA   NA
```