

HW4_Convergence&STAN

N. Barrus

2024-02-14

Purpose:

The purpose of this markdown document is to work through Homework 4 in Dr. Babcock's Bayesian Statistics Course at the University of Miami. Homework 4 deals with assessing convergence and introducing the class to STAN.

Please note, that STAN package compiles the code in C++ before using it in R. This means the first time the code runs it is a bit slower, but subsequent attempts are much quicker. Stan uses a Hamiltonian MCMC which greatly speeds up convergence, but must be used with numerical data.

General Start to Code

```
rm(list = ls())

#####github#####
#note, only needed after 90 days from 1/16/2024

# usethis::create_github_token()
# gitcreds::gitcreds_set()

#####check for r updates#####
#note, updateing may take some time so plan accordingly

#require(installr)

#check.for.updates.R()

#updateR() #only if needed

#####check for package updates#####
#note, updateing may take some time so plan accordingly

#old.packages()

# update.packages() #make the decision to the update the packages
```

Load packages

```
library(tidyverse)
library(R2jags)
library(rstan)
library(ggmcmc)
library(here)
theme_set(theme_bw(base_size=15))
```

Data

The data consists of counts of the number of trees in 30 equal sized quadrats.

```
Y <- c(11, 3, 7, 6, 2, 36, 14, 9, 2, 10, 2, 7, 3, 1, 0, 0, 0, 1,
5, 0, 2, 11, 5, 3, 0, 3, 3, 27, 0, 11)

tree.counts <- list(Y = Y,
                    N = length(Y))

tree.counts

## $Y
## [1] 11 3 7 6 2 36 14 9 2 10 2 7 3 1 0 0 0 1 5 0 2 11 5
## [24] 3 0 3 3 27 0 11
##
## $N
## [1] 30
```

Problem 1:

Fit this data in JAGS, using the Poisson distribution. Call the parameter lambda, where $\lambda = \text{mean} = \text{variance}$, and use the lognormal distribution for the prior. Print the parameter summaries, and show the traceplot and density. Is the model converged? *Write and Run The Jags model*

```
#write the model

write(("model{

#parameter priors

lamda ~ dlnorm(0.0, 1.0E-6)

#likelihood

for(i in 1:N) {
Y[i] ~ dpois(lamda)
}

}"), here("JAGS_mods", "HW3-Q1-Poisson.txt"))

#run the model
```

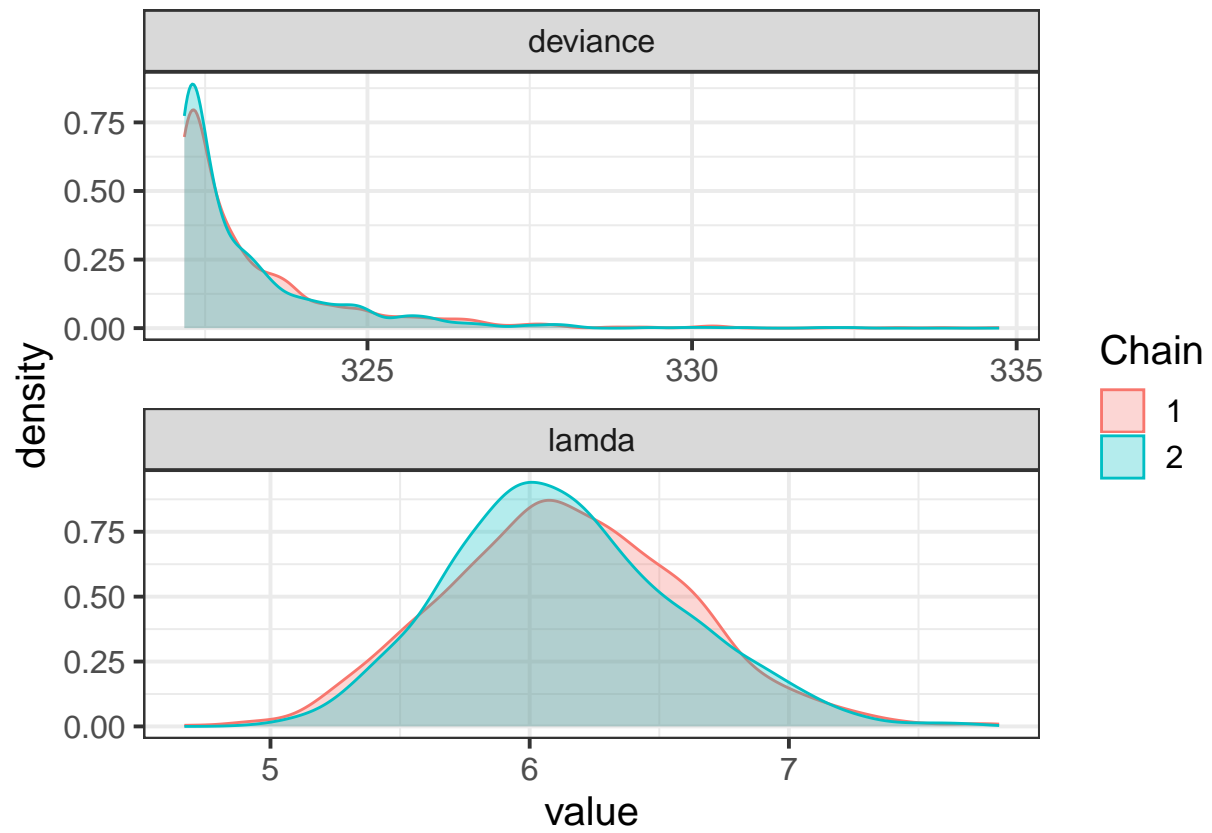
```
treemod.HW4.Q1.poiss <- jags(data = tree.counts,
  parameters.to.save = c("lamda"),
  n.chains = 2,
  n.burnin = 1000,
  n.iter = 20000,
  model.file = here("JAGS_mods", "HW3-Q1-Poisson.txt")
)
```

Convergence Diagnostics

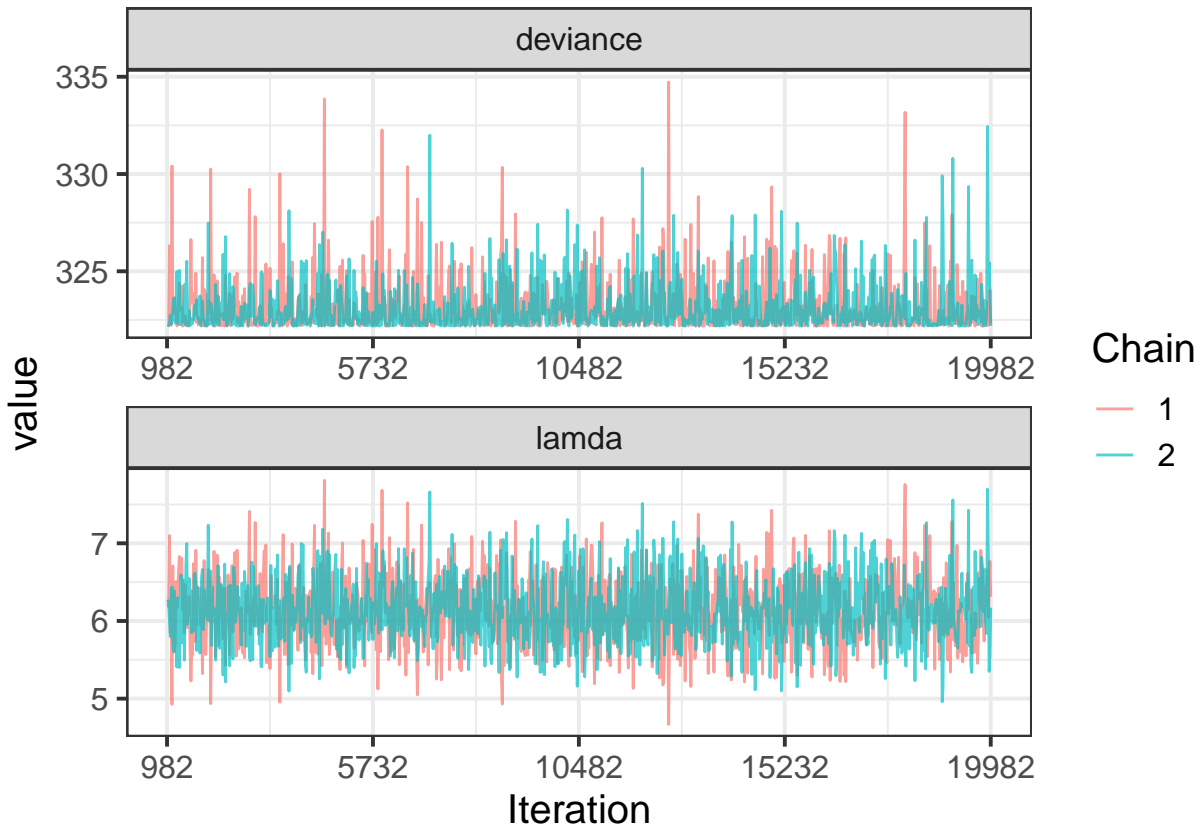
```
treemod.HW4.Q1.poiss$BUGSoutput
```

```
## Inference for Bugs model at "C:/Users/nbarr/OneDrive/Pictures/Documents/FIU/Classes/BayesianStatistics/
## 2 chains, each with 20000 iterations (first 1000 discarded), n.thin = 19
## n.sims = 2000 iterations saved
##          mean sd 2.5% 25% 50% 75% 97.5% Rhat n.eff
## deviance 323.2 1.4 322.2 322.3 322.6 323.5 326.8 1 730
## lamda      6.1 0.4 5.3 5.8 6.1 6.4 7.1 1 2000
##
## 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 = 1.0 and DIC = 324.1
## DIC is an estimate of expected predictive error (lower deviance is better).
```

```
ggs.treemod.HW4.Q1.poiss <- ggs(as.mcmc(treemod.HW4.Q1.poiss))
ggs_density(ggs.treemod.HW4.Q1.poiss)
```



```
ggs_traceplot(ggs.treemod.HW4.Q1.pois)
```



The model has adequately converged, because the model neff is greater than 400, the rhats are close to 1, the densities are relatively smooth, there appears to be little autocorrelation in the traceplots, and the plots show little difference between chains.

Problem 2:

Fit this data in STAN, using the Poisson distribution. The likelihood for a Poisson distribution in STAN is called `poisson`, with the same parameter as in JAGS. Print the parameter summaries, and show the traceplot and density. Is the model converged? Are the results similar to the JAGS results? *run the model the STAN model*

```
treemod.HW4.Q2.poiss <- stan(file = here("STAN_docs", "HW4_Q2_poisson.stan"),
                             data = tree.counts)
```

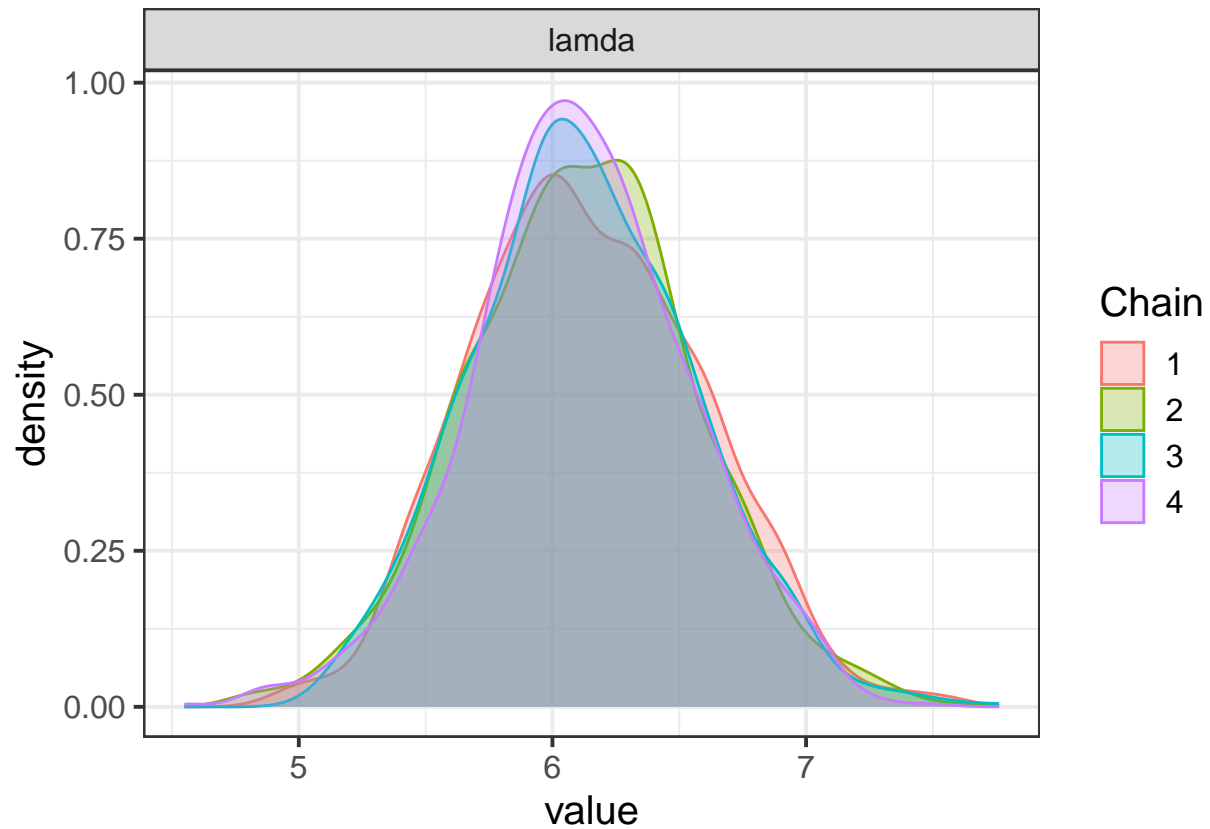
Convergence Diagnostics

```
print(treemod.HW4.Q2.poiss)
```

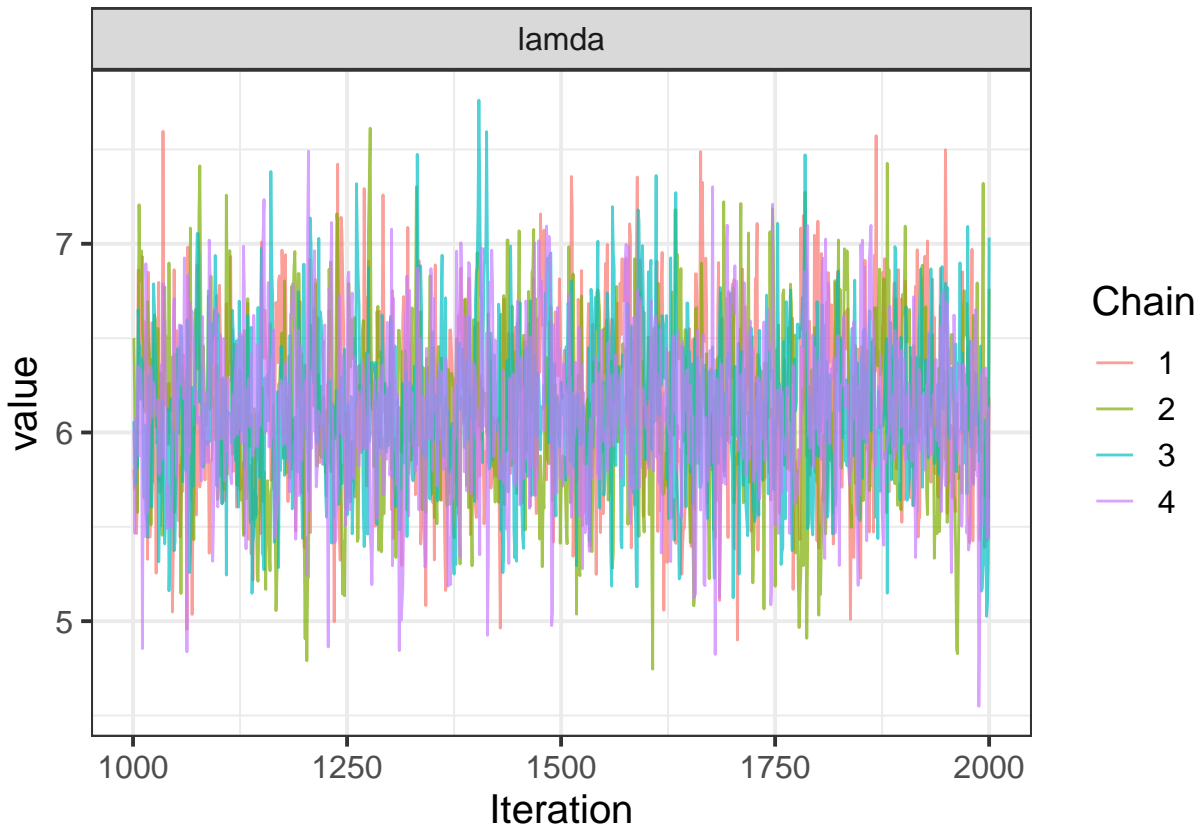
```
## Inference for Stan model: anon_model.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##          mean se_mean  sd  2.5%   25%   50%   75%  97.5% n_eff Rhat
## lamda    6.12    0.01 0.44   5.29   5.83   6.11   6.41   6.99  1546    1
```

```
## lp__ 148.33    0.01 0.69 146.38 148.18 148.59 148.76 148.81 2089    1
##
## Samples were drawn using NUTS(diag_e) at Thu Feb 15 13:04:43 2024.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

```
ggs.treemod.HW4.Q2.poiss <- ggs(treemod.HW4.Q2.poiss)
ggs_density(ggs.treemod.HW4.Q2.poiss)
```



```
ggs_traceplot(ggs.treemod.HW4.Q2.poiss)
```



The model has adequately converged, because the model neff is greater than 400, the rhats are close to 1, the densities are relatively smooth, there appears to be little autocorrelation in the traceplots, and the plots show little difference between chains.

The JAGS and the STAN methods showed similar results.

Problem 3:

Fit the data using the original probability (p), number of successes (r) parameterization of the negative binomial using JAGS. Use a uniform prior for p and a lognormal for r . Calculate the mean and variance as estimated parameters. Print the summary statistics, density plot, traceplot, pairs plot for p and r , and autocorrelation plot. What convergence issues can you diagnose? Is the variance larger than the mean? *Write and Run The Jags model*

```
write(("model{
#parameter priors
p ~ dunif(0,1) #prior for propotion
r ~ dlnorm(0.0, 1.0E-6) #prior for number of successes
#likelihood
for(i in 1:N) {
Y[i] ~ dnegbin(p,r)
}
```

```

#estimate parameters
m <- r *(1-p)/p
v <- m+m*m/r
}), here("JAGS_mods", "HW3-Q3-NegBinom.txt"))

treemod.HW4.Q3.negbin <- jags(data = tree.counts,
  parameters.to.save = c("p", "r", "m", "v"),
  n.chains = 2,
  n.burnin = 1000,
  n.iter = 20000,
  model.file = here("JAGS_mods", "HW3-Q3-NegBinom.txt"))

```

Convergence Diagnostics

```
treemod.HW4.Q3.negbin$BUGSoutput
```

```

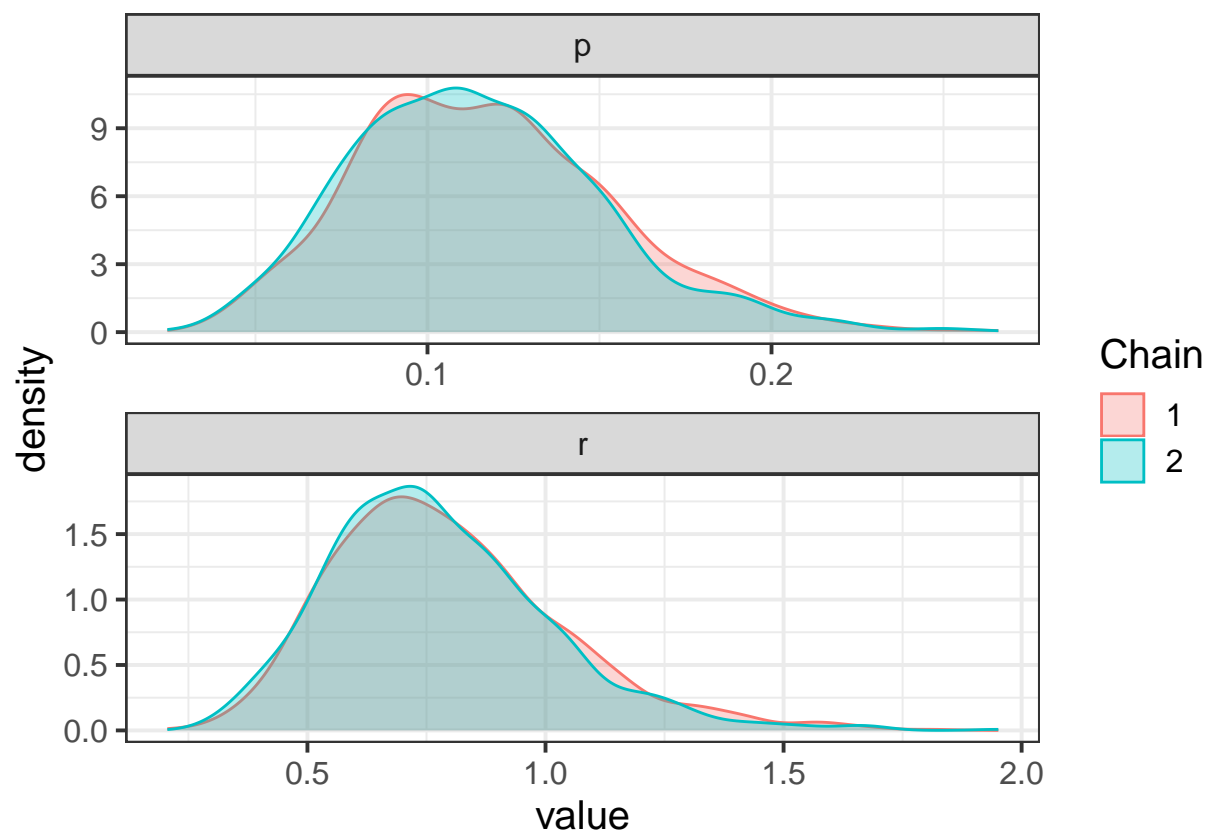
## Inference for Bugs model at "C:/Users/nbarr/OneDrive/Pictures/Documents/FIU/Classes/BayesianStatistics/
## 2 chains, each with 20000 iterations (first 1000 discarded), n.thin = 19
## n.sims = 2000 iterations saved
##          mean   sd  2.5%  25%   50%   75%  97.5% Rhat n.eff
## deviance 174.4  2.0 172.5 173.0 173.8 175.1 179.6    1  2000
## m         6.2  1.4   3.9   5.1   6.0   7.0   9.3    1  2000
## p         0.1  0.0   0.1   0.1   0.1   0.1   0.2    1   890
## r         0.8  0.2   0.4   0.6   0.8   0.9   1.3    1   740
## v        62.4 37.9  22.8  38.6  52.6  75.1 153.5    1  1600
##
## 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 = 1.9 and DIC = 176.3
## DIC is an estimate of expected predictive error (lower deviance is better).

```

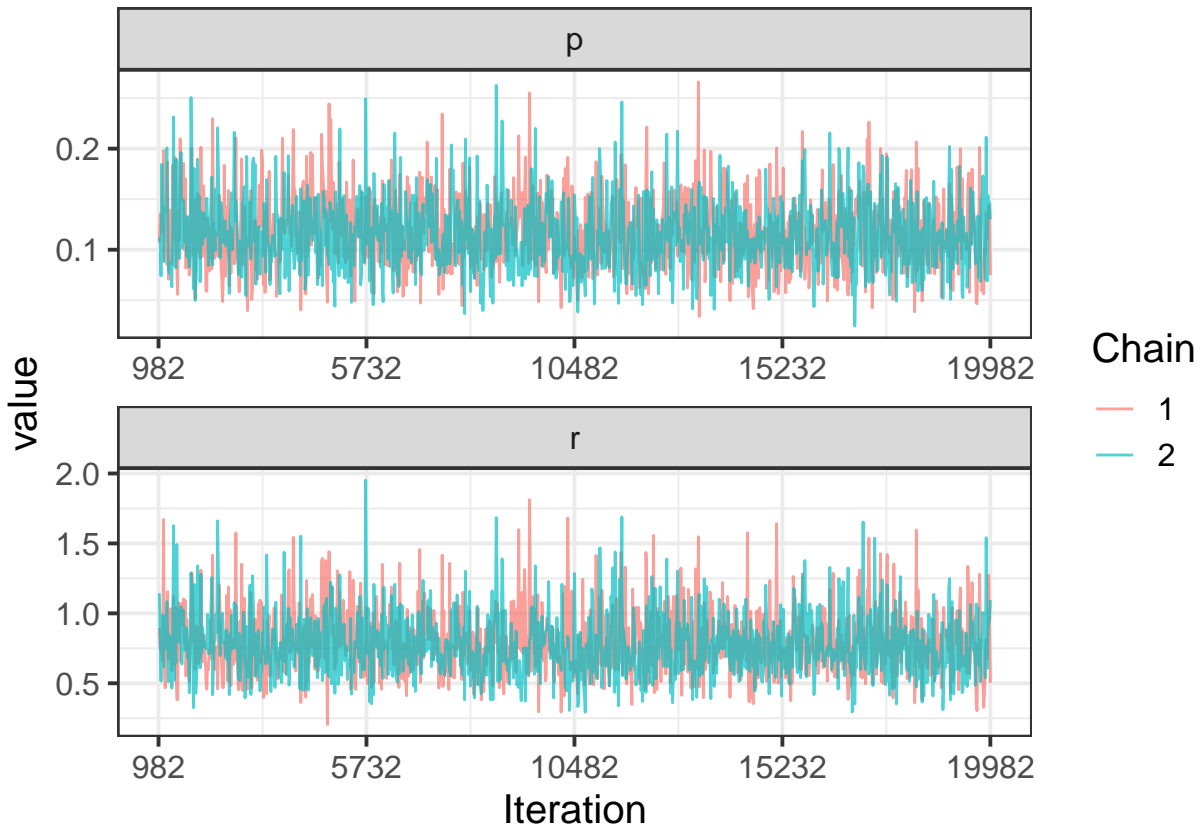
```

ggs.treemod.HW4.Q3.negbin <- ggs(as.mcmc(treemod.HW4.Q3.negbin)) |>
  filter(Parameter == "p" | Parameter == "r")
ggs_density(ggs.treemod.HW4.Q3.negbin)

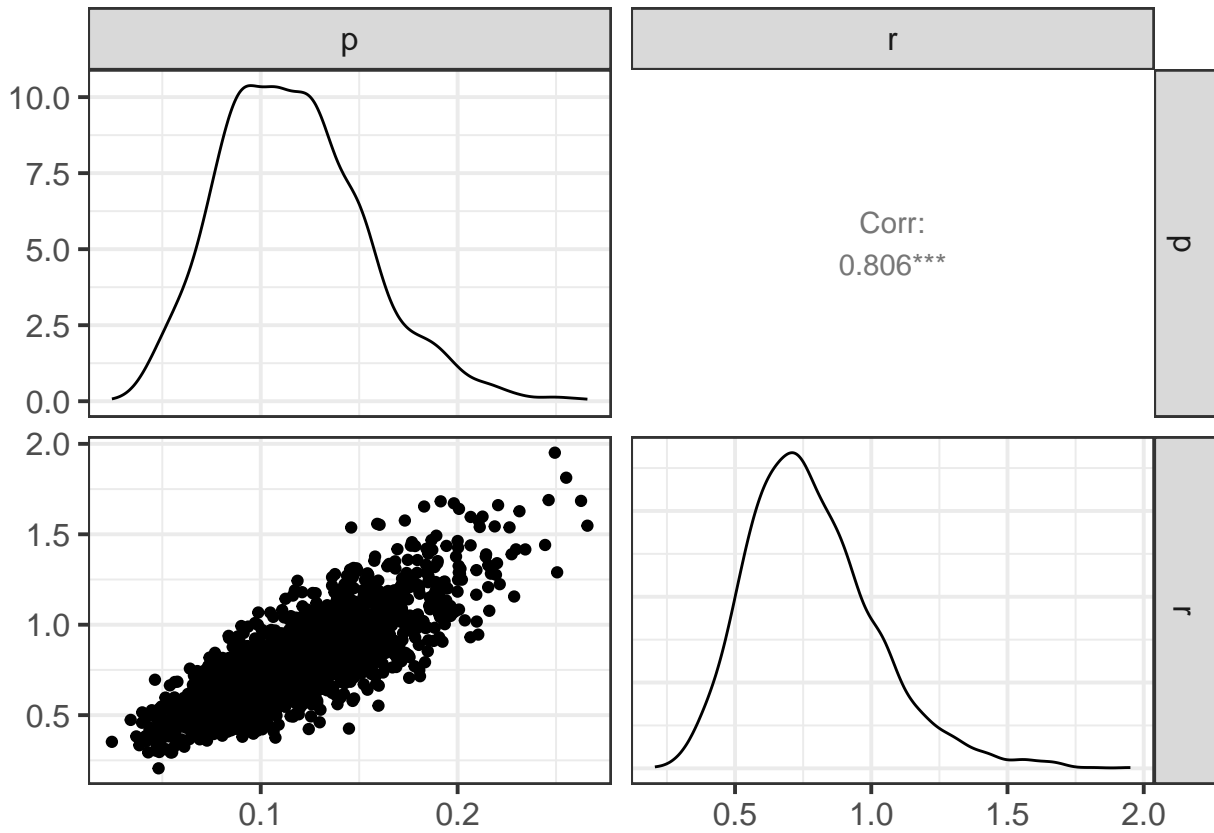
```

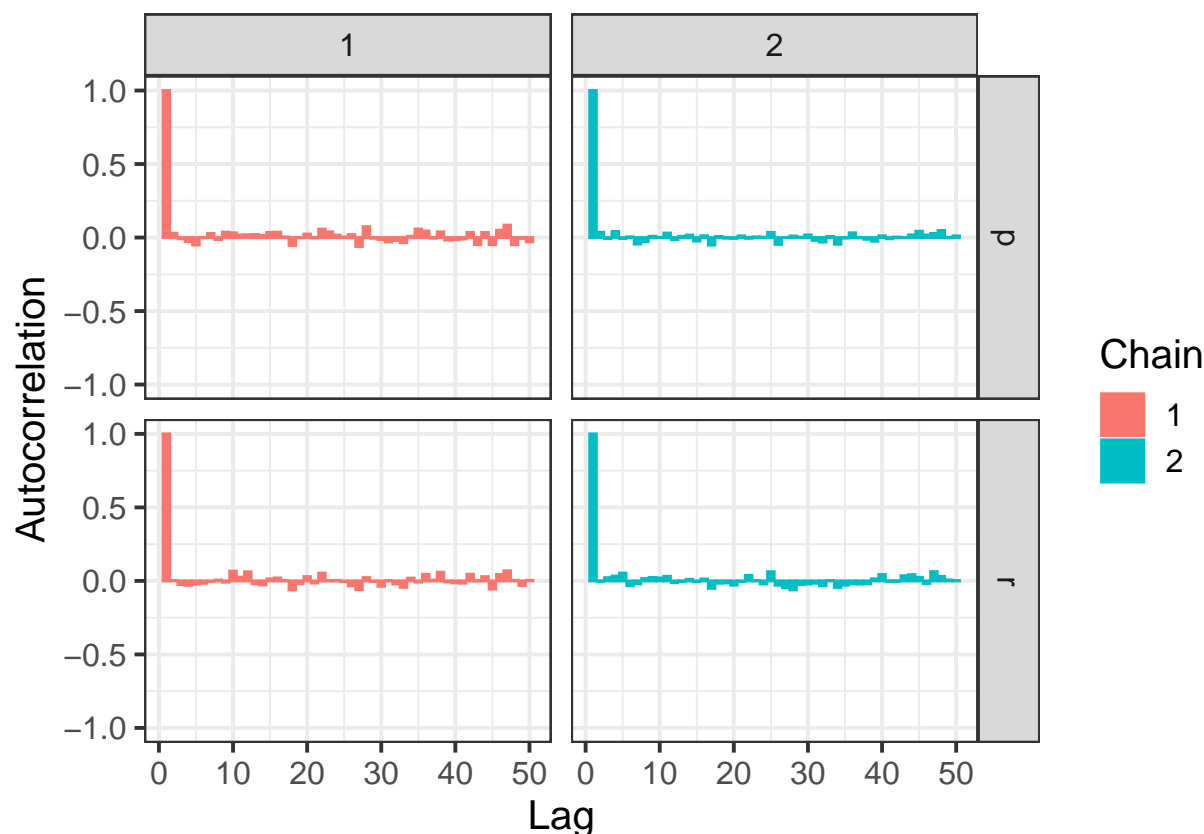
```
ggs_traceplot(ggs.treemod.HW4.Q3.negbin)
```



```
ggs_pairs(ggs.treemod.HW4.Q3.negbin)
```



```
ggs_autocorrelation(ggs.treemod.HW4.Q3.negbin)
```



I can't spot any glaring convergence issues, in that rhats are close to one, effective sample size is > 400 for all parameters, autocorrelation looks the same for each draw and is low with very minor instance of going below zero, the traceplot is relatively random across the traces and the chains are similar, and the posterior densities are relatively smooth and similar across chains. I guess there is relatively high correlation between the p and r parameters which could produce convergence problems, but the diagnostics suggest otherwise.

The variance (61.8) is larger than the mean (6.2).

Problem 4:

Fit the same model to the data using STAN, with the same parameterization, and using the `neg_binomial_2` for the likelihood. For the r parameter, don't give it an upper limit> ie define it as: `real<lower=0> r;` Estimate the variance and mean as either transformed parameters or estimated quantities. Look at the same convergence diagnostics, plus `stan_diag` for divergences. How does the convergence compare to the previous model in JAGS? *run the stan model*

```
treemod.HW4.Q4.negbinom <- stan(file = here("STAN_docs", "HW4_Q4_negbinom-2.stan"),
                                data = tree.counts)
```

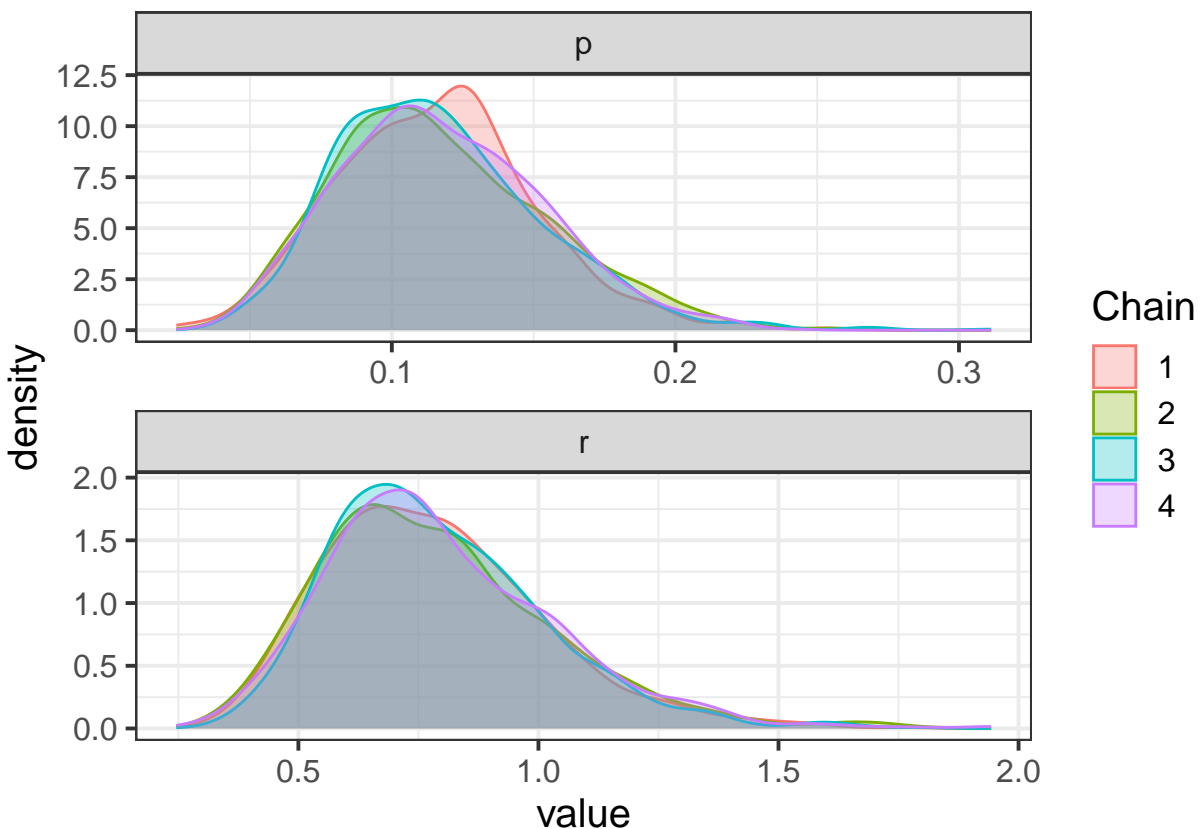
```
## Warning in readLines(file, warn = TRUE): incomplete final line found on
## 'C:\Users\nbarr\OneDrive\Pictures\Documents\FIU\Classes\BayesianStatistic_Course\BayesianStatistic_R\
```

Convergence Diagnostics

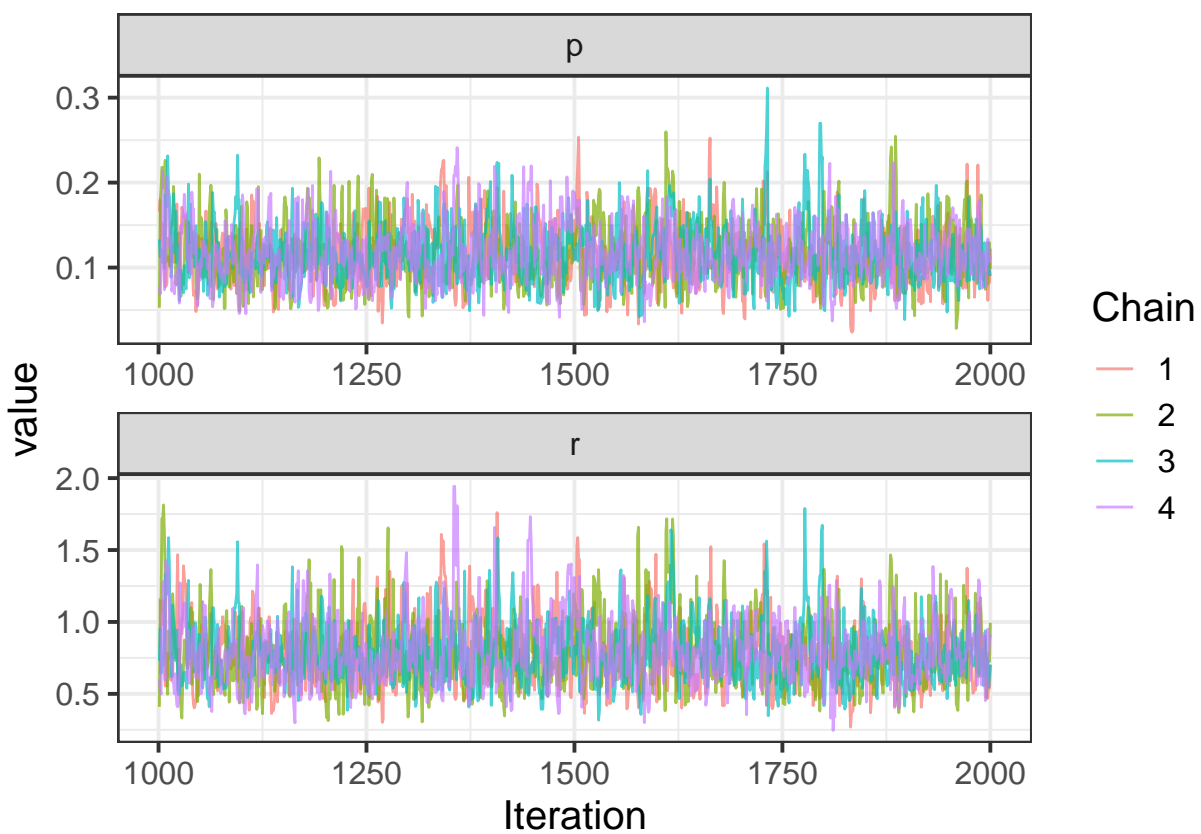
```
print(treemod.HW4.Q4.negbinom)
```

```
## Inference for Stan model: anon_model.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##      mean se_mean   sd  2.5%  25%  50%  75%  97.5% n_eff Rhat
## r      0.79   0.01  0.23  0.42  0.62  0.76  0.93  1.33 1195   1
## p      0.12   0.00  0.04  0.06  0.09  0.11  0.14  0.19 1299   1
## m      6.16   0.02  1.45  3.91  5.15  5.96  6.99  9.35 3456   1
## v     61.50   0.90 38.58 23.20 37.90 52.06 73.27 156.75 1846   1
## lp__ -90.43   0.03  1.00 -93.01 -90.81 -90.13 -89.73 -89.47 1557   1
##
## Samples were drawn using NUTS(diag_e) at Thu Feb 15 13:05:20 2024.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

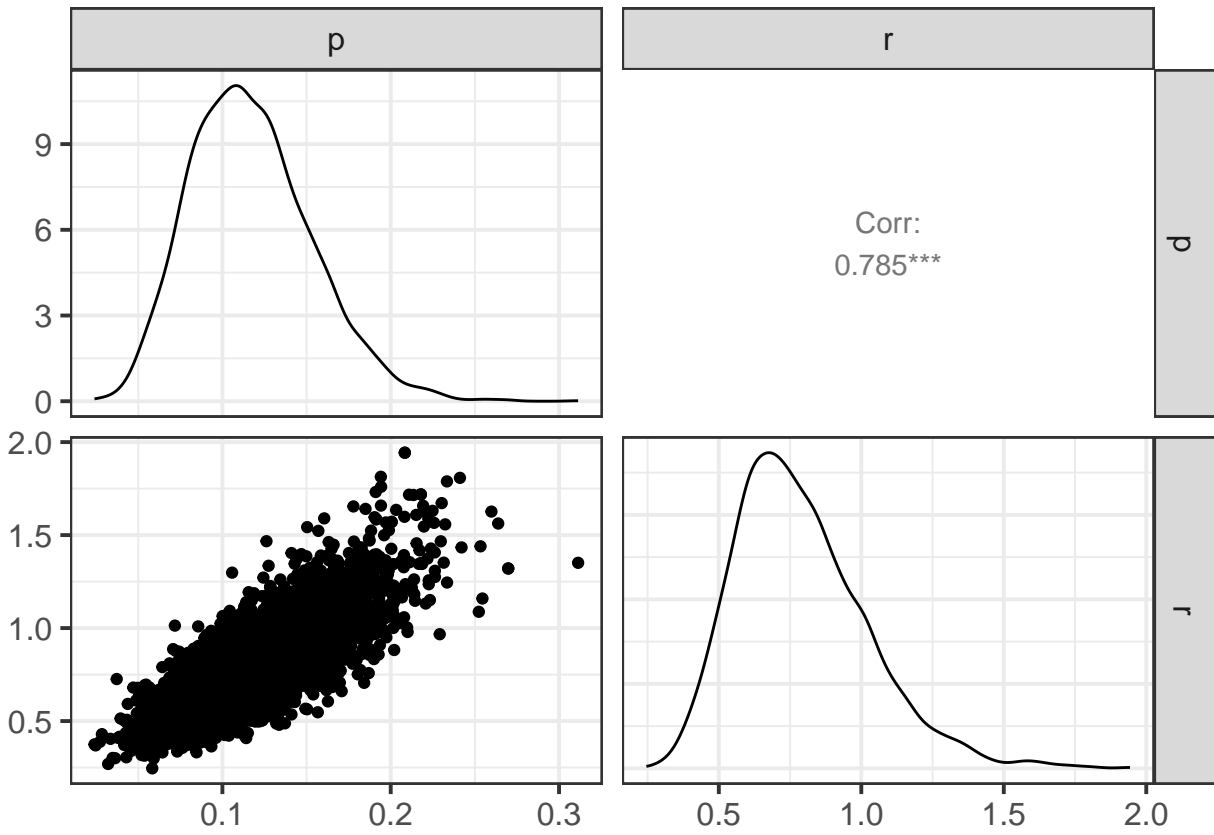
```
ggs.treemod.HW4.Q4.negbin <- ggs(treemod.HW4.Q4.negbinom) |>
  filter(Parameter == "p" | Parameter == "r")
ggs_density(ggs.treemod.HW4.Q4.negbin)
```



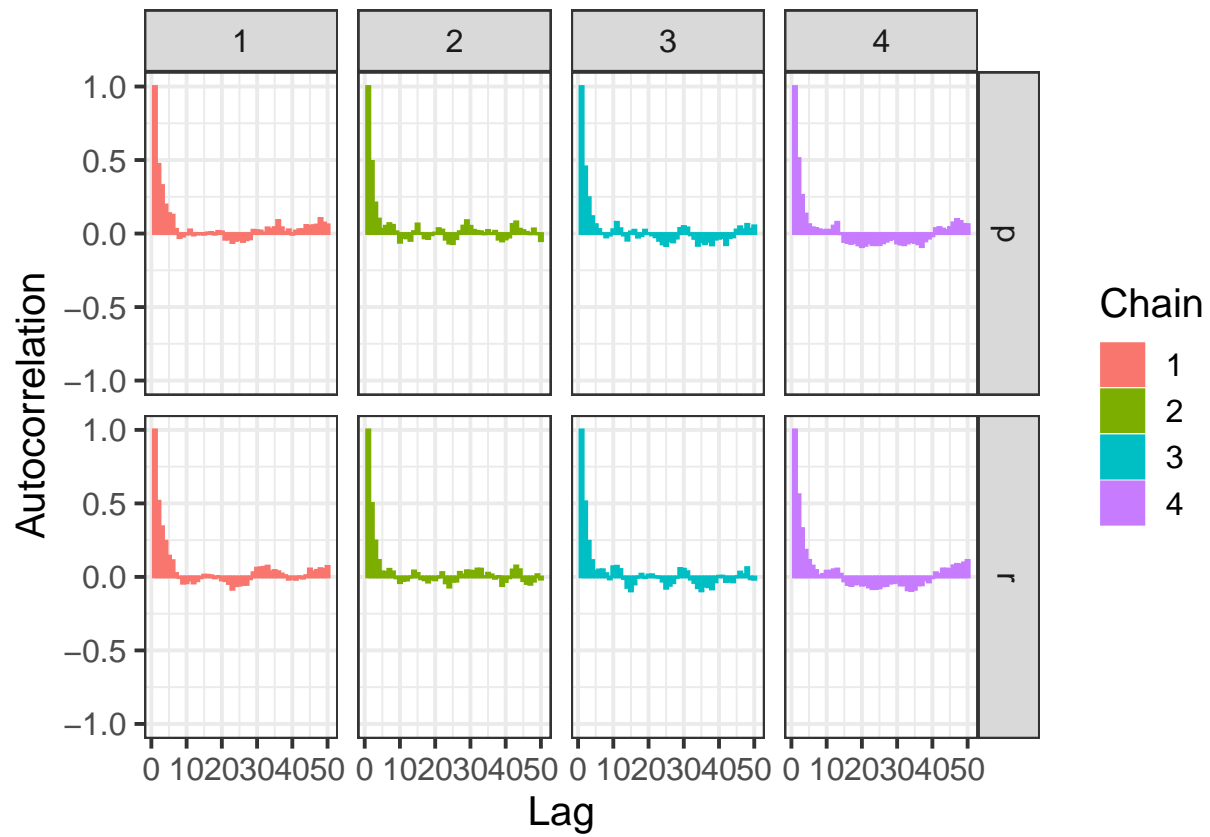
```
ggs_traceplot(ggs.treemod.HW4.Q4.negbin)
```



```
ggs_pairs(ggs.treemod.HW4.Q4.negbin)
```

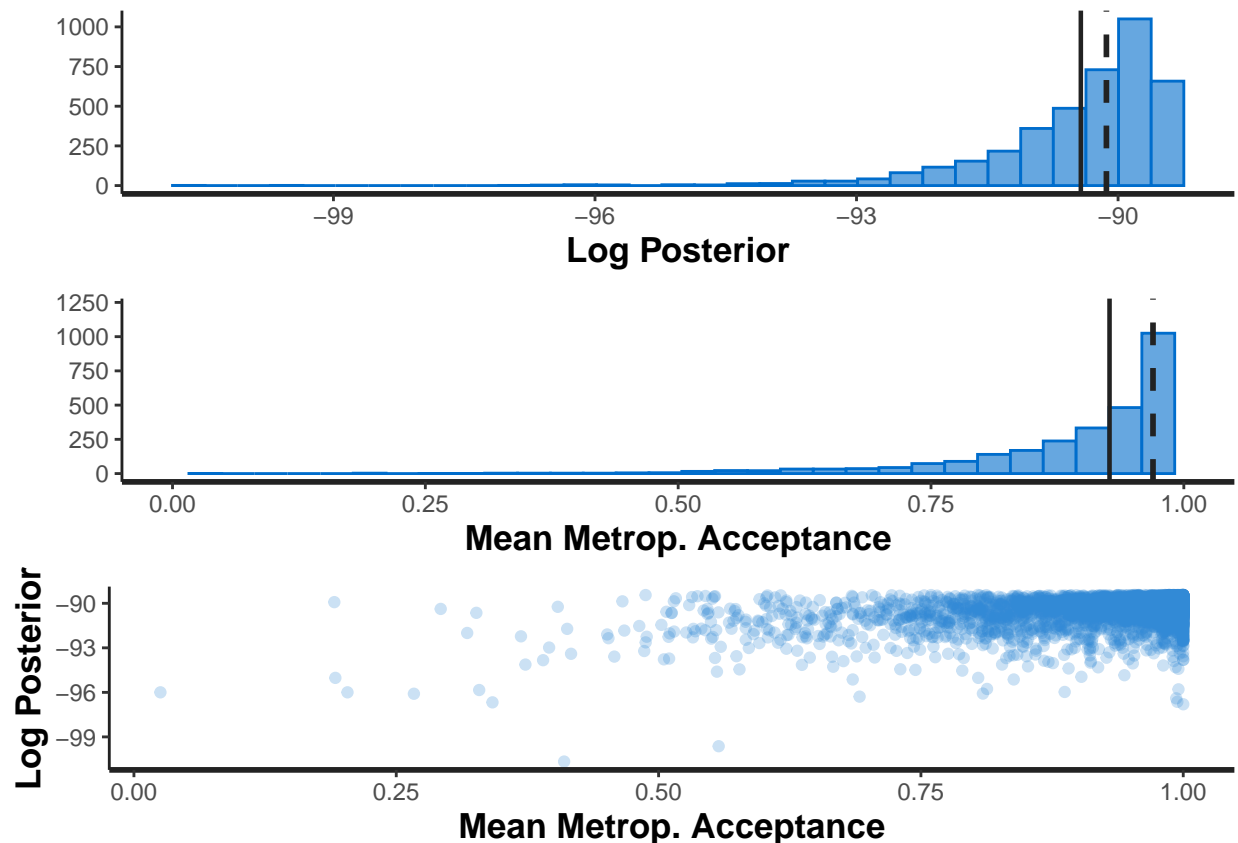


```
ggs_autocorrelation(ggs.treemod.HW4.Q4.negbin)
```



```
stan_diag(treemod.HW4.Q4.negbinom)
```

```
## Warning: Removed 2 rows containing missing values ('geom_bar()').
```

The convergence is similar to that in Q3, the effective sample sizes are consistently larger than the lowest effective sample size in Q3 (720). The autocorrelation was about the same, the rhats were close to 1, the trace plots were about the same as in Q3. The STAN diag plots didn't show any divergences.

Problem 5:

Fit the data using the mean (m) as a parameter rather than probability (p), still using r as a parameter. Calculated mean and variance as estimated quantities. Give r an upper bound of 10 and use less informative priors. Estimate the variance as an estimated quantity. Look at the same convergence diagnostics. How does the convergence compare to the previous model in STAN? Do the mean and variance look similar to parts 3 and 4? *run the stan model*

```
treemod.HW4.Q5.negbinom <- stan(file = here("STAN_docs", "HW4_Q5_negbinom-2.stan"),
                                data = tree.counts)
```

```
## Warning in readLines(file, warn = TRUE): incomplete final line found on
## 'C:\Users\nbarr\OneDrive\Pictures\Documents\FIU\Classes\BayesianStatistic_Course\BayesianStatistic_R\
```

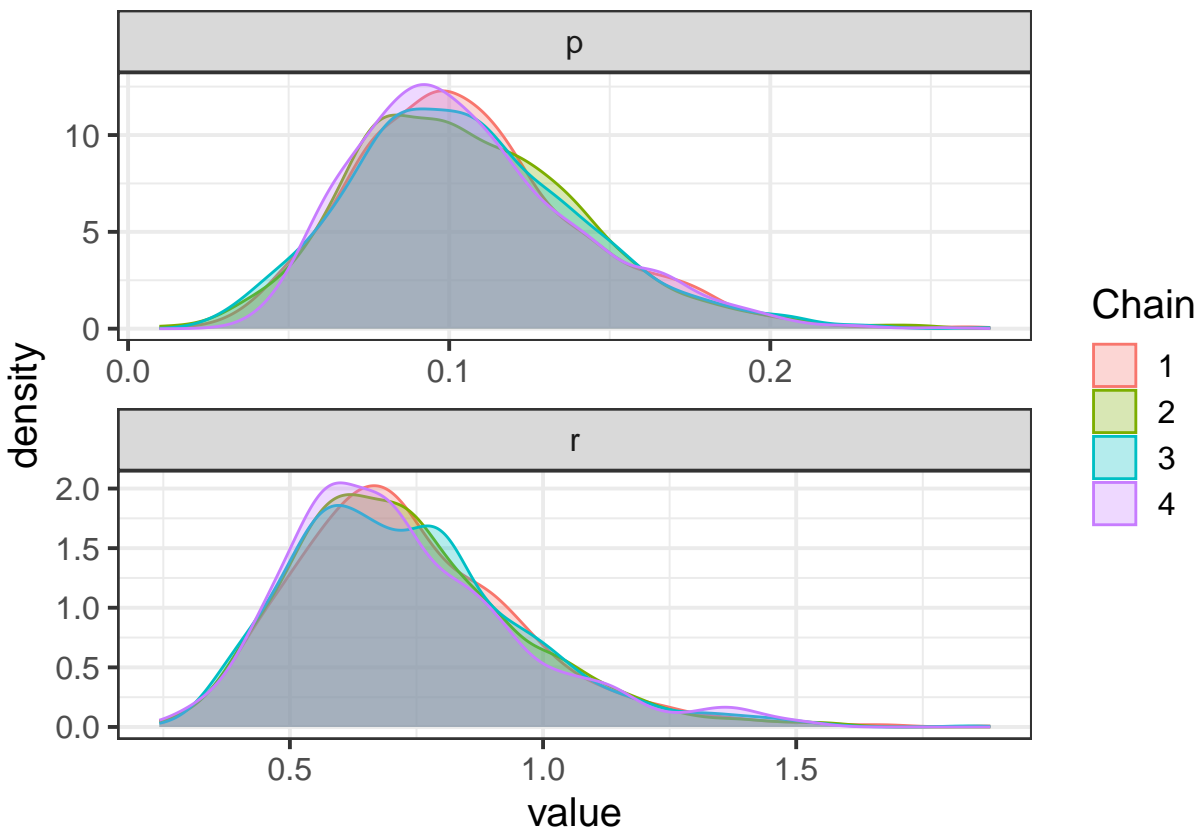
Convergence Diagnostics

```
print(treemod.HW4.Q5.negbinom)
```

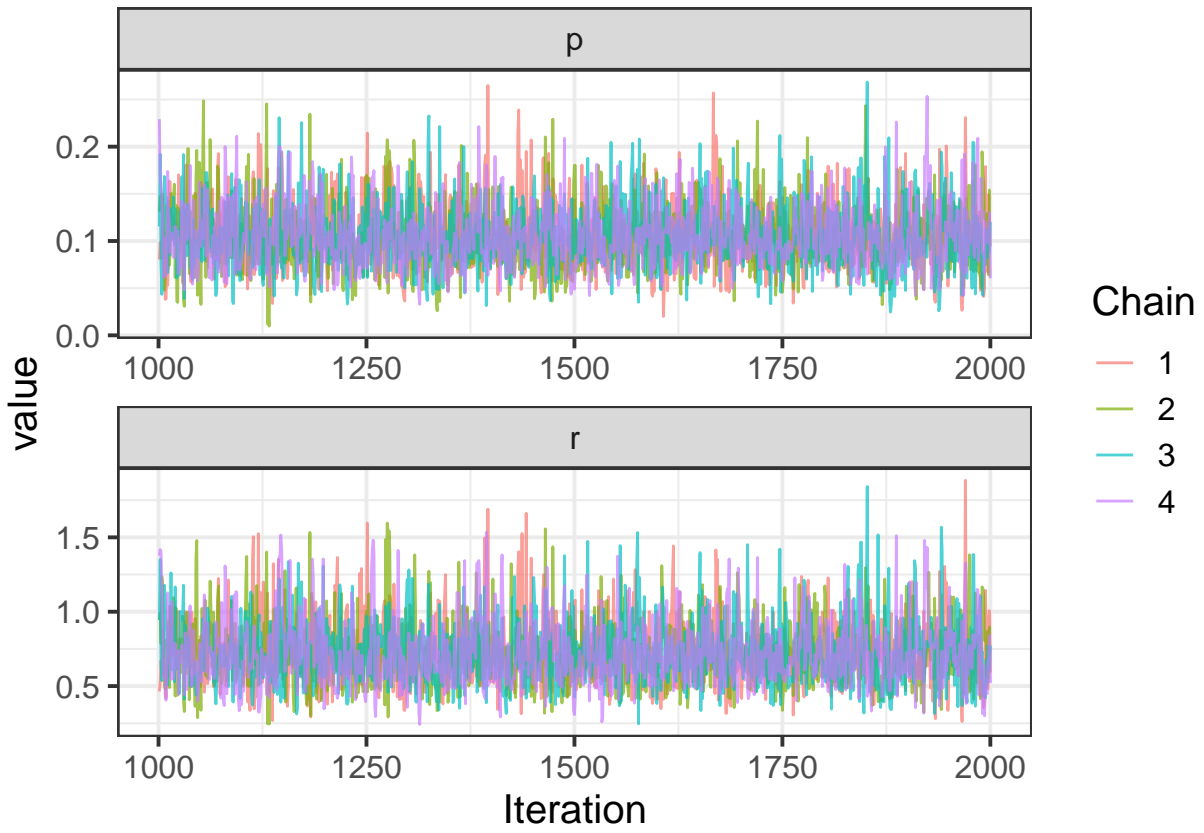
```
## Inference for Stan model: anon_model.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
```

```
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##      mean se_mean      sd  2.5%   25%   50%   75%  97.5% n_eff Rhat
## r      0.73    0.00  0.22  0.38   0.57   0.70   0.85   1.27 2823   1
## m      6.43    0.03  1.67  3.92   5.28   6.23   7.29  10.24 2861   1
## v     74.12    1.56 69.47 24.67  43.21  59.85  87.42 200.72 1983   1
## p      0.11    0.00  0.04  0.05   0.08   0.10   0.13   0.19 3047   1
## lp__ -89.17    0.03  1.07 -91.89 -89.55 -88.82 -88.42 -88.15 1786   1
##
## Samples were drawn using NUTS(diag_e) at Thu Feb 15 13:05:54 2024.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

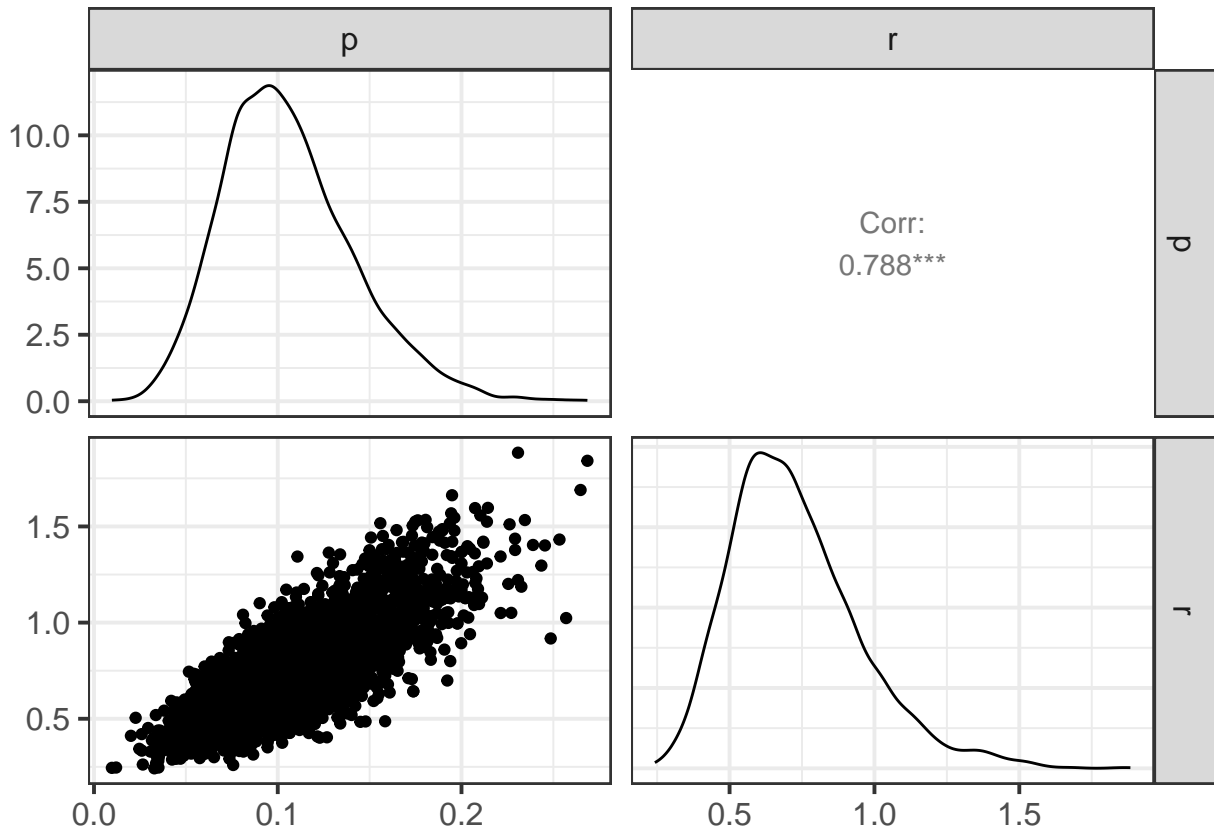
```
ggs.treemod.HW4.Q5.negbin <- ggs(treemod.HW4.Q5.negbinom) |>
  filter(Parameter == "p" | Parameter == "r")
ggs_density(ggs.treemod.HW4.Q5.negbin)
```



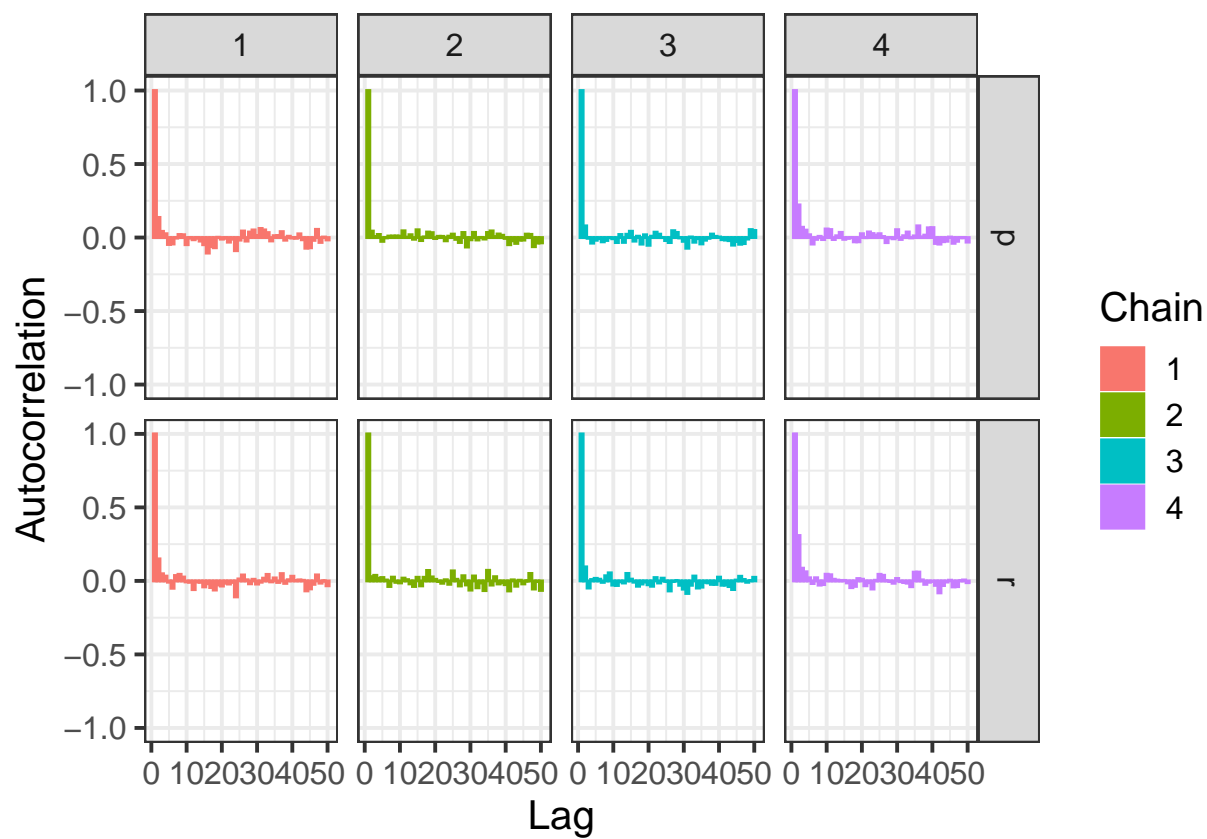
```
ggs_traceplot(ggs.treemod.HW4.Q5.negbin)
```



```
ggs_pairs(ggs.treemod.HW4.Q5.negbin)
```

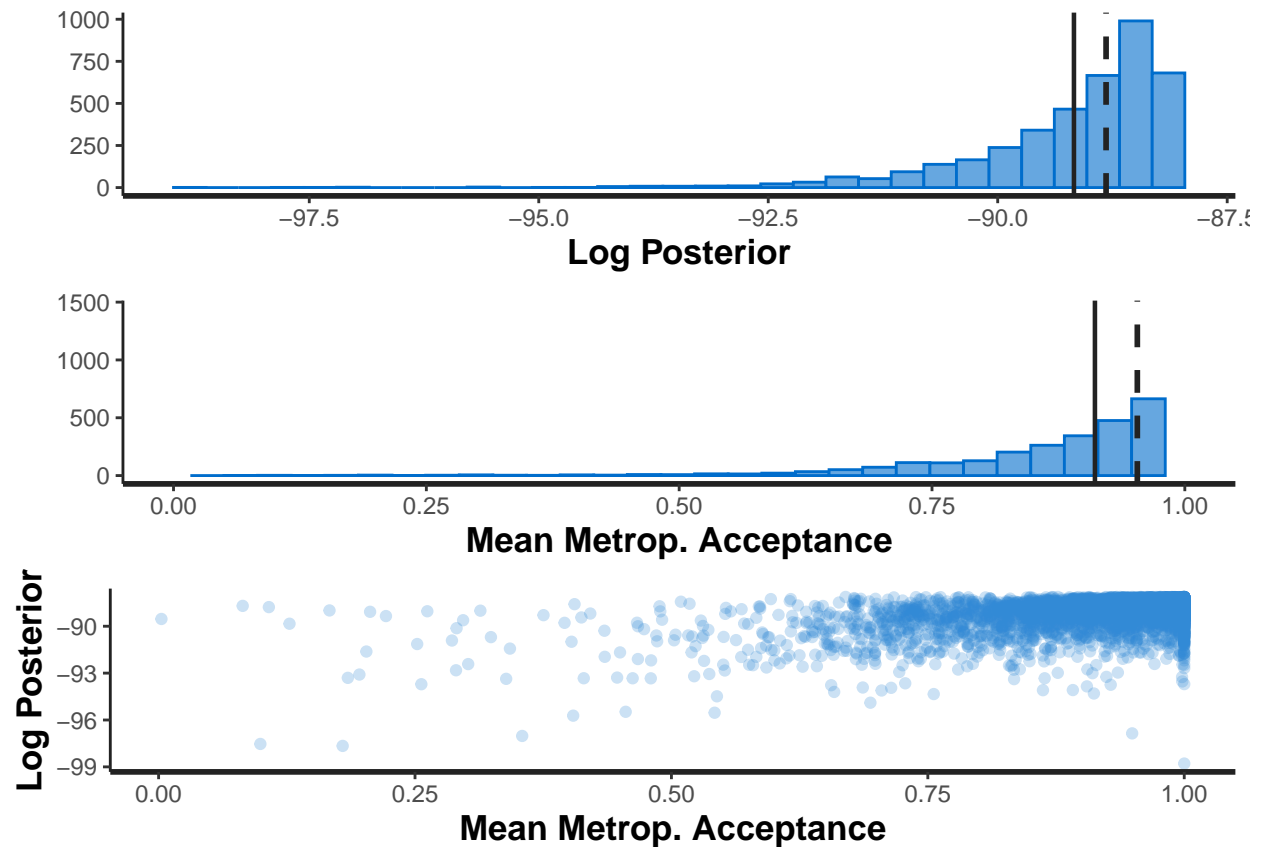


```
ggs_autocorrelation(ggs.treemod.HW4.Q5.negbin)
```



```
stan_diag(treemod.HW4.Q5.negbinom)
```

```
## Warning: Removed 2 rows containing missing values ('geom_bar()').
```



The convergence is similar to that in Q4.

comparison between Q3, Q4, Q5

```
print(treemod.HW4.Q3.negbin)
```

```
## Inference for Bugs model at "C:/Users/nbarr/OneDrive/Pictures/Documents/FIU/Classes/BayesianStatistic
## 2 chains, each with 20000 iterations (first 1000 discarded), n.thin = 19
## n.sims = 2000 iterations saved
##      mu.vect sd.vect   2.5%   25%   50%   75%  97.5%  Rhat
## m      6.167   1.406   3.923   5.146   6.009   6.966   9.304  1.001
## p      0.116   0.037   0.053   0.089   0.113   0.138   0.198  1.002
## r      0.783   0.236   0.408   0.616   0.751   0.916   1.331  1.003
## v     62.434  37.919  22.774  38.568  52.560  75.094 153.456  1.001
## deviance 174.378   1.965 172.471 172.996 173.784 175.051 179.634  1.000
##      n.eff
## m      2000
## p      890
## r      740
## v     1600
## deviance 2000
##
## 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 = 1.9 and DIC = 176.3
## DIC is an estimate of expected predictive error (lower deviance is better).
```

```
print(treemod.HW4.Q4.negbinom)
```

```
## Inference for Stan model: anon_model.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##      mean se_mean      sd  2.5%   25%   50%   75%  97.5% n_eff Rhat
## r      0.79     0.01  0.23  0.42  0.62  0.76  0.93  1.33 1195   1
## p      0.12     0.00  0.04  0.06  0.09  0.11  0.14  0.19 1299   1
## m      6.16     0.02  1.45  3.91  5.15  5.96  6.99  9.35 3456   1
## v     61.50     0.90 38.58 23.20 37.90 52.06 73.27 156.75 1846   1
## lp__ -90.43     0.03  1.00 -93.01 -90.81 -90.13 -89.73 -89.47 1557   1
##
## Samples were drawn using NUTS(diag_e) at Thu Feb 15 13:05:20 2024.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

```
print(treemod.HW4.Q5.negbinom)
```

```
## Inference for Stan model: anon_model.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##      mean se_mean      sd  2.5%   25%   50%   75%  97.5% n_eff Rhat
## r      0.73     0.00  0.22  0.38  0.57  0.70  0.85  1.27 2823   1
## m      6.43     0.03  1.67  3.92  5.28  6.23  7.29 10.24 2861   1
## v     74.12     1.56 69.47 24.67 43.21 59.85 87.42 200.72 1983   1
## p      0.11     0.00  0.04  0.05  0.08  0.10  0.13  0.19 3047   1
## lp__ -89.17     0.03  1.07 -91.89 -89.55 -88.82 -88.42 -88.15 1786   1
##
## Samples were drawn using NUTS(diag_e) at Thu Feb 15 13:05:54 2024.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

The neff in this model appear to be on average larger than the those in Q4. But the Rhats are lower, the autocorrelation plot is a little less but similar, and the traceplots look good. The stan_diag plots showed no divergences.

The mean and the variances look similar to those in parts 3 and 4.