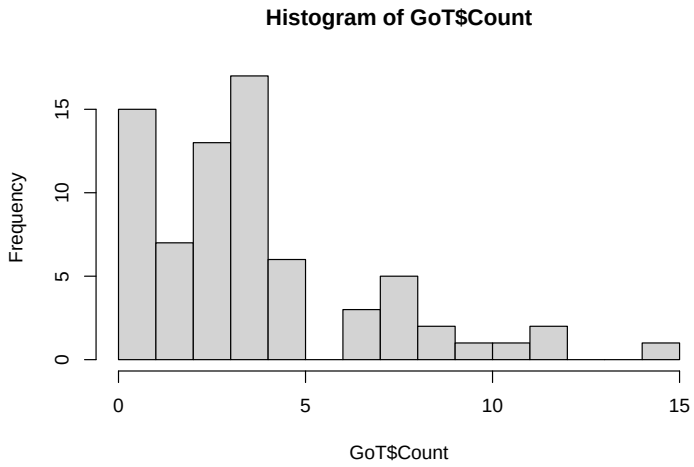
STA 602 Lab 3

Yicheng Shen

19 September, 2022

Exercise 1

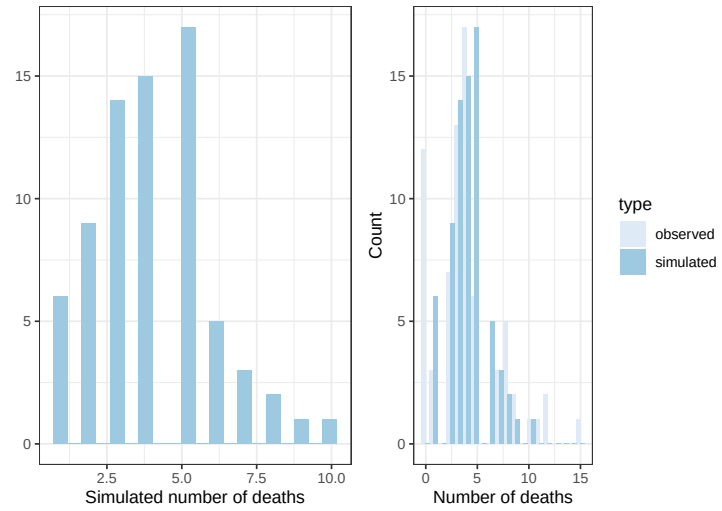
```
GoT <- readxl::read_xlsx("GoT_Deaths.xlsx", col_names = T)
hist(GoT$Count, breaks = 15)
```



Exercise 2

```
y <- GoT$Count
n <- length(y)

# Finish: obtain empirical mean
ybar <- mean(y)
sim_dat <- rpois(n, ybar)
a <- qplot(sim_dat, bins = 20, xlab = "Simulated number of deaths", fill = I("#9ecae1"))

df <- data.frame(rbind(data.frame(y, "observed") %>% dplyr::rename(data = 1, type = 2),
            data.frame(sim_dat, "simulated") %>% dplyr::rename(data = 1, type = 2)))
b <- ggplot(df, aes(x = data, fill = type))+
```

```
  geom_histogram(position = "dodge", bins = 20) +
  scale_fill_brewer()  +
  labs(x = "Number of deaths", y = "Count")

grid.arrange(a,b, nrow =1)
```

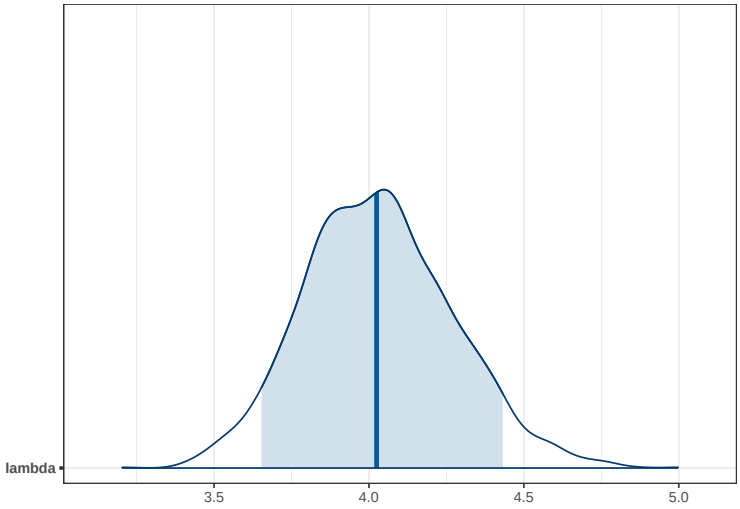



```
set.seed(8848)
stan_dat <- list(y = y, N = n)
fit <- stan("lab-03-poisson-simple.stan", data = stan_dat, refresh = 0, chains = 2)
```

```
## Trying to compile a simple C file
```

```
lambda_draws <- as.matrix(fit, pars = "lambda")
```

```
mcmc_areas(lambda_draws, prob = 0.90)
```



```r
mean(GoT$Count) # sample mean
```

```
## [1] 4.013699
```

```r
print(fit, pars = "lambda")
```

```
## Inference for Stan model: lab-03-poisson-simple.
## 2 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=2000.
##
##        mean se_mean   sd 2.5%  25%  50%  75% 97.5% n_eff Rhat
## lambda 4.03    0.01 0.24 3.58 3.86 4.02 4.19  4.53   727    1
##
## Samples were drawn using NUTS(diag_e) at Mon Sep 19 10:29:26 2022.
## 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).
```

Answer: The sample mean from data is 4.013699, which is quite coherent with the posterior mean. The posterior mean is still higher than the sample mean because it is a compromise between prior and data, and the prior mean is 5.

Exercise 3

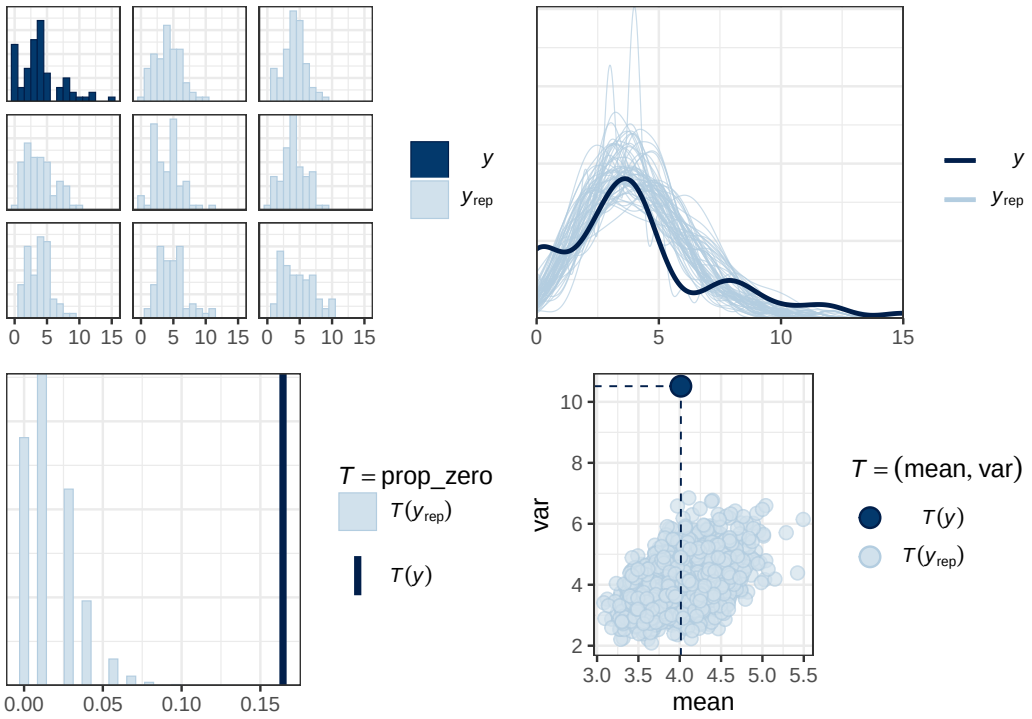
```r
y_rep <- as.matrix(fit, pars = "y_rep")

set.seed(8848)
posterior_sample_draw <- list()
for (i in seq_along(lambda_draws))
{
  posterior_sample_draw[[i]] <- rpois(n, lambda_draws[i])
}
y_rep <- matrix(unlist(posterior_sample_draw),
                nrow = length(lambda_draws), ncol = n, byrow = T)
```

Exercise 4

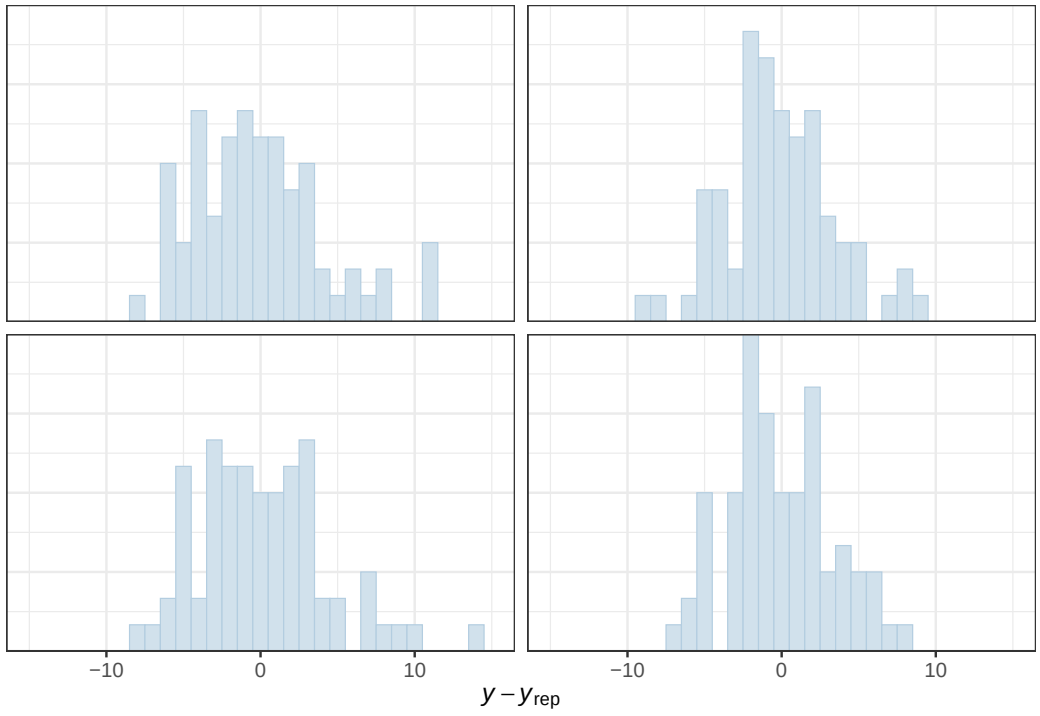
```r
a<-ppc_hist(y, y_rep[1:8, ], binwidth = 1)
b<-ppc_dens_overlay(y, y_rep[1:60, ])

prop_zero <- function(x){
  mean(x == 0)
}
prop_zero(y) ##  0.164
```

```
## [1] 0.1643836
```

```
c<-ppc_stat(y, y_rep, stat = "prop_zero")
d<-ppc_stat_2d(y, y_rep, stat = c("mean", "var"))
grid.arrange(a,b,c,d,nrow=2)
```




```
ppc_error_hist(y, y_rep[1:4, ], binwidth = 1) + xlim(-15, 15)
```



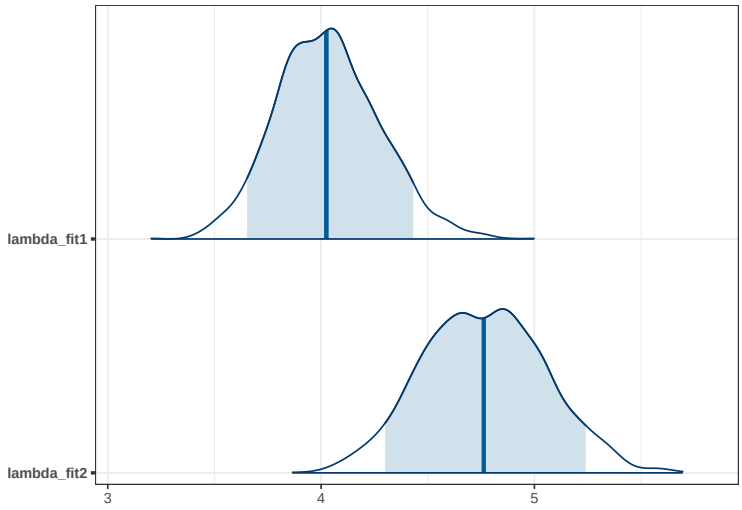
Answer: It is clear that the model does not fit the data perfectly since the observed data seems to be very a unusual datapoint among the posterior sample draws.

Exercise 5

```
set.seed(8848)
fit2 <- stan("lab-03-poisson-hurdle.stan", data = stan_dat, refresh = 0, chains = 2)
```

```
## Trying to compile a simple C file
```

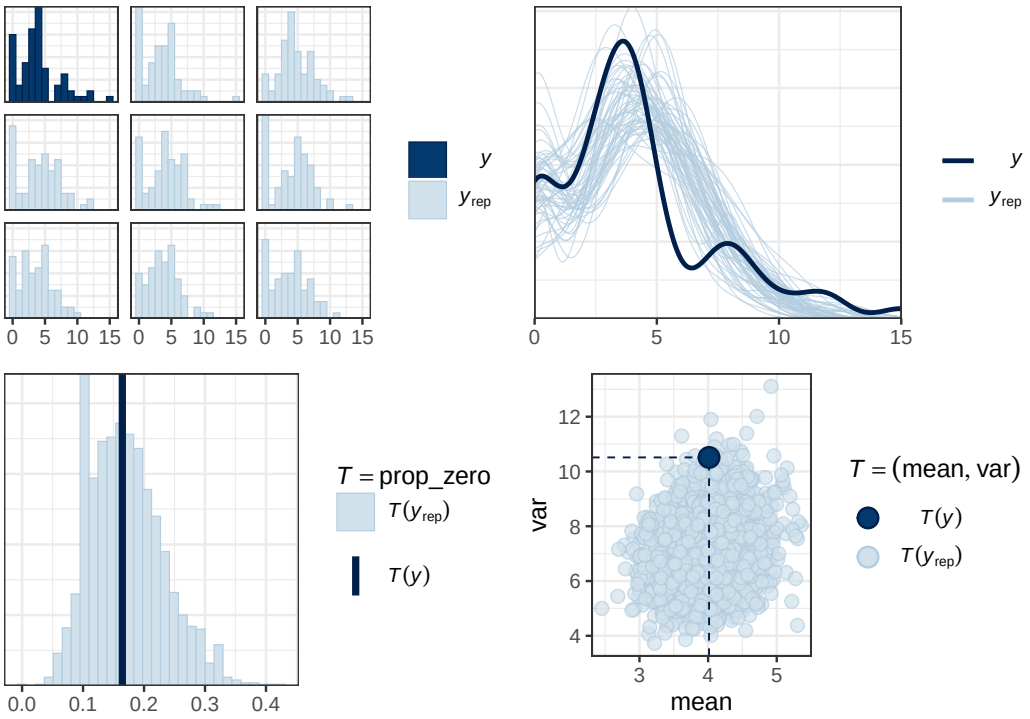
```
# Extract the sampled values for lambda, and store them in a variable called lambda_draws2:
lambda_draws2 <- as.matrix(fit2, pars = "lambda")
# Compare
lambdas <- cbind(lambda_fit1 = lambda_draws[, 1],
                 lambda_fit2 = lambda_draws2[, 1])
# Shade 90% interval
mcmc_areas(lambdas, prob = 0.9)
```



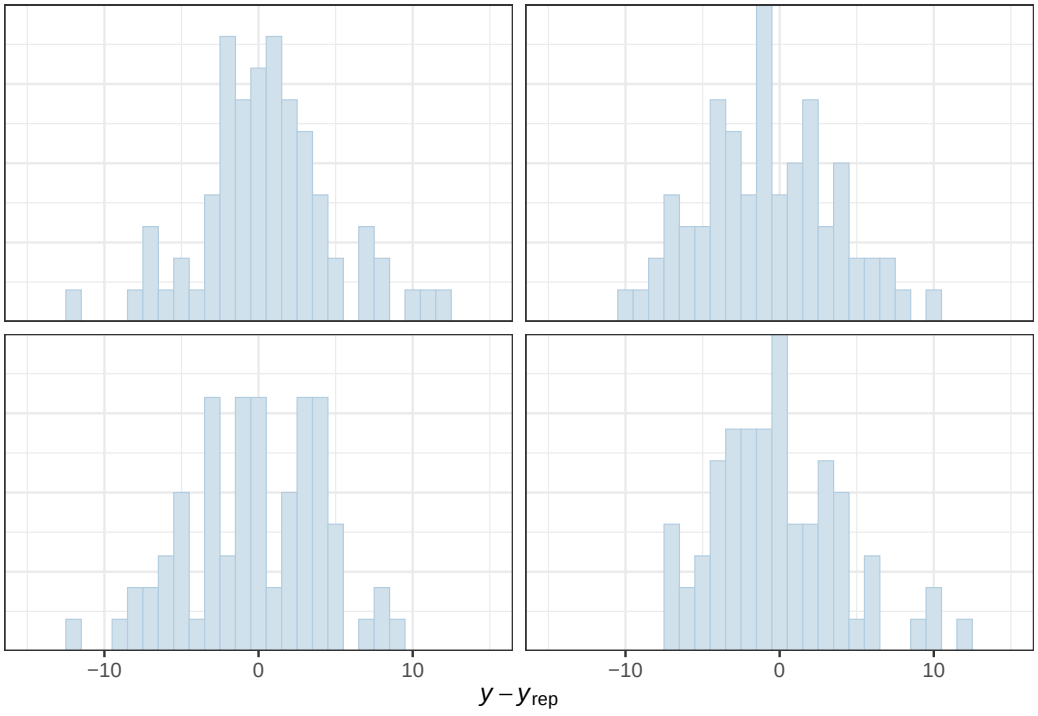

```
y_rep2 <- as.matrix(fit2, pars = "y_rep")
```

Exercise 6

```
a<-ppc_hist(y, y_rep2[1:8, ], binwidth = 1)
b<-ppc_dens_overlay(y, y_rep2[1:60, ])
c<-ppc_stat(y, y_rep2, stat = "prop_zero")
d<-ppc_stat_2d(y, y_rep2, stat = c("mean", "var"))
grid.arrange(a,b,c,d,nrow=2)
```




```
ppc_error_hist(y, y_rep2[1:4, ], binwidth = 1) + xlim(-15, 15)
```



Answer: This model is a much better fit since the observed data lies reasonably within the posterior draws.

Exercise 7

```
log_lik1 <- extract_log_lik(fit, merge_chains = FALSE)
r_eff1 <- relative_eff(exp(log_lik1))
(loo1 <- loo(log_lik1, r_eff = r_eff1))
```

```
##
## Computed from 2000 by 73 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo   -203.3 14.6
## p_loo         2.7  0.6
## looic       406.6 29.2
## ------
## Monte Carlo SE of elpd_loo is 0.1.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```

```
log_lik2 <- extract_log_lik(fit2, merge_chains = FALSE)
r_eff2 <- relative_eff(exp(log_lik2))
(loo2 <- loo(log_lik2, r_eff = r_eff2))
```

```
##
## Computed from 2000 by 73 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo   -183.4 10.9
## p_loo         2.9  0.5
## looic       366.7 21.7
## ------
## Monte Carlo SE of elpd_loo is 0.0.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```

```
loo_compare(loo1, loo2)
```

```
##        elpd_diff se_diff
## model2   0.0       0.0
## model1 -19.9       8.5
```

Answer: `model2` is a better fit here considering its smaller prediction error.

Exercise 8

Answer: If our model is a reasonable fit, then we should see that the observed data lies reasonably among the simulated draws from posterior.

PPC is important because it helps us check whether our model could generate something similar to the observed data. If not, then our model is does not fit well.

Exercise 9

Answer: The second model should be a good fit for the data. Based on PPC, the observed data is a reasonably close to the posterior draws. It is also shown that the model fits data with smaller LOOCV error.

Exercise 10

Answer: A single LOOCV error may not be enough to confidently choose a model (we might want to compare it with some other model's error). However, it is still useful in a way that we know whether the observed data is far from the fitted model's draws or not.
