STA 602 Lab

Will Tirone

19 September, 2022

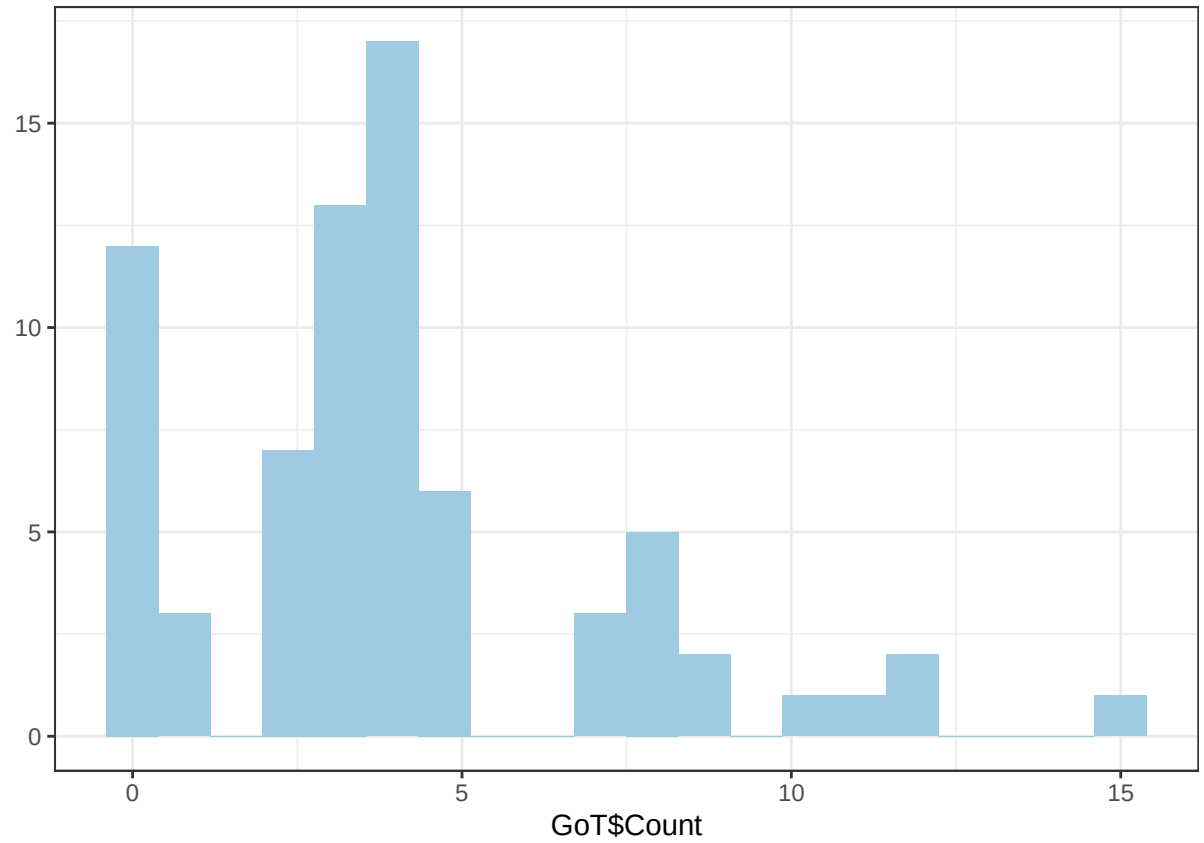
$Y_1, ..., Y_n|\lambda \sim iidPOIS(\lambda)\lambda \sim GAM(a,b)p(\lambda|Y_1, ..., Y_n) \propto P(Y_1, ..., Y_n|\lambda)p(\lambda)p(\lambda|Y_1, ..., Y_n) \sim GAM(a+\sum Y_i, n+b) = (shap$

Posterior Predictive: use this to simulate new data

1) sim λ_i following posterior distribution. $ Y_1,...,Y_n) \sim GAMMA$
2) simulate $\tilde{Y_i}, ..., \tilde{Y_n} \sim P(Y_1, ..., Y_n|\tilde{\lambda})$ (this is just a dataset)

Exercise 1

```
GoT <- read_excel("GoT_Deaths.xlsx", col_names = T)
qplot(GoT$Count, bins = 20,fill = I("#9ecae1"))
```



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

Exercise 2

The mean for our sample is about 4, which is pretty close to the simulated data. I'd guess they're slightly different just from some kind of sampling variability.

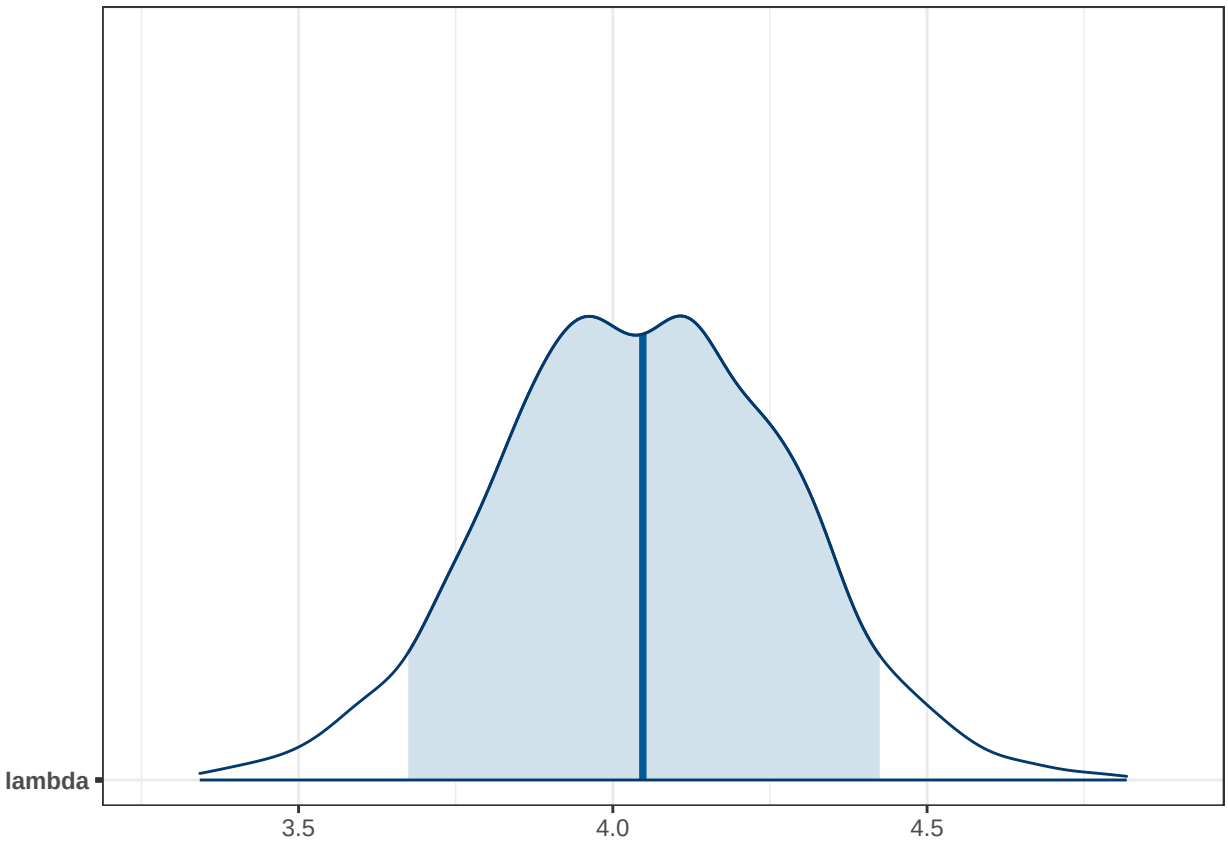
```
stan_dat <- list(y = y, N = n)
fit <- stan("lab-03-poisson-simple.stan", data = stan_dat, refresh = 0, chains = 2)
```

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

```
## Running /usr/lib64/R/bin/R CMD SHLIB foo.c
## gcc -m64 -I"/usr/include/R" -DNDEBUG   -I"/usr/lib64/R/library/Rcpp/include/"  -I"/usr/lib64/R/libra
## In file included from /usr/lib64/R/library/RcppEigen/include/Eigen/Dense:1,
##                  from /usr/lib64/R/library/StanHeaders/include/stan/math/prim/mat/fun/Eigen.hpp:13,
##                  from <command-line>:
## /usr/lib64/R/library/RcppEigen/include/Eigen/Core:82:12: fatal error: new: No such file or directory
##    82 |   #include <new>
##       |            ^~~~~
## compilation terminated.
## make: *** [/usr/lib64/R/etc/Makeconf:168: foo.o] Error 1
```

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

mcmc_areas(lambda_draws, prob = 0.90)
```



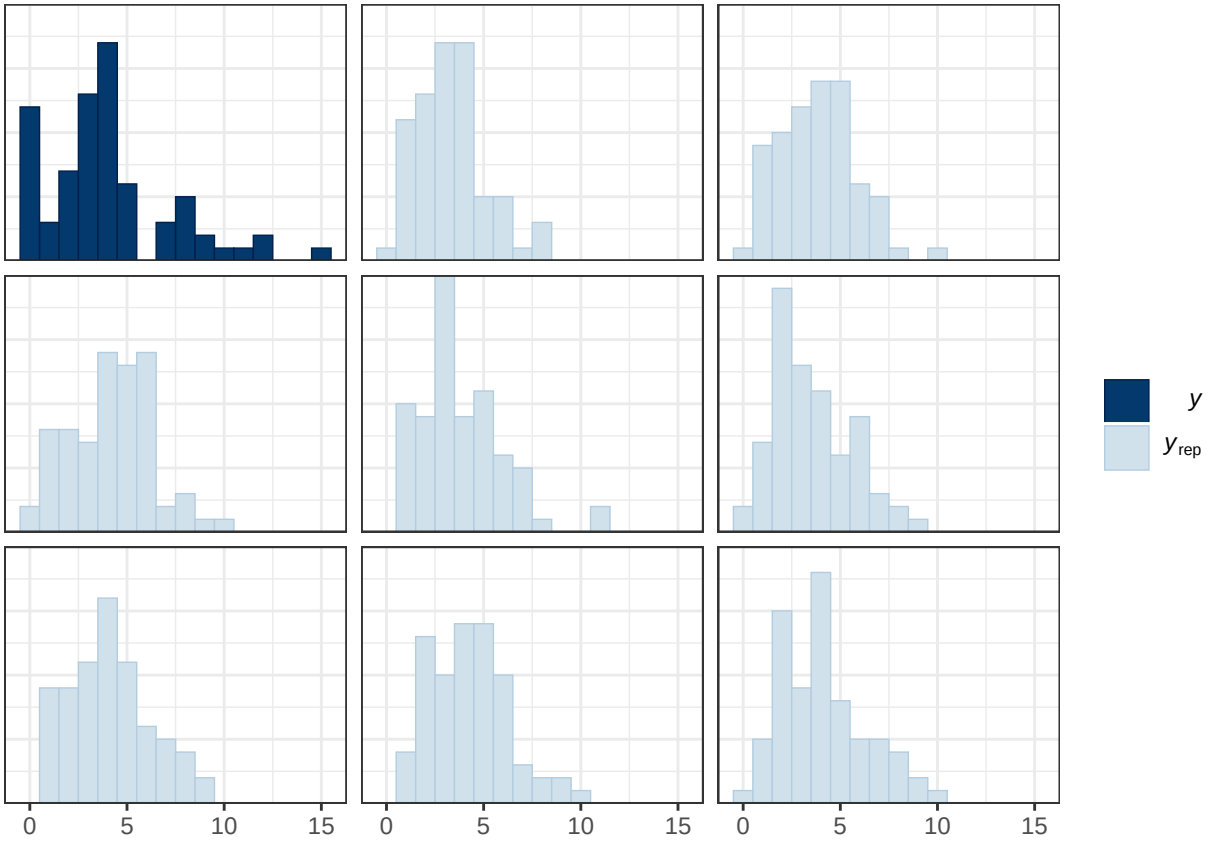
```
print(fit, pars = "lambda")
```

```
## 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 4.05    0.01 0.23  3.6 3.89 4.05 4.21   4.5   789    1
##
## Samples were drawn using NUTS(diag_e) at Mon Sep 19 14:32:25 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).
```

Exercise 3

```
yrep_matrix = matrix(NA,nrow=nrow(lambda_draws),ncol=length(y))
for (i in 1:2000){
  yrep_matrix[i,] = rpois(length(y),lambda_draws[i,])
}

ppc_hist(y, yrep_matrix[1:8, ], binwidth = 1)
```

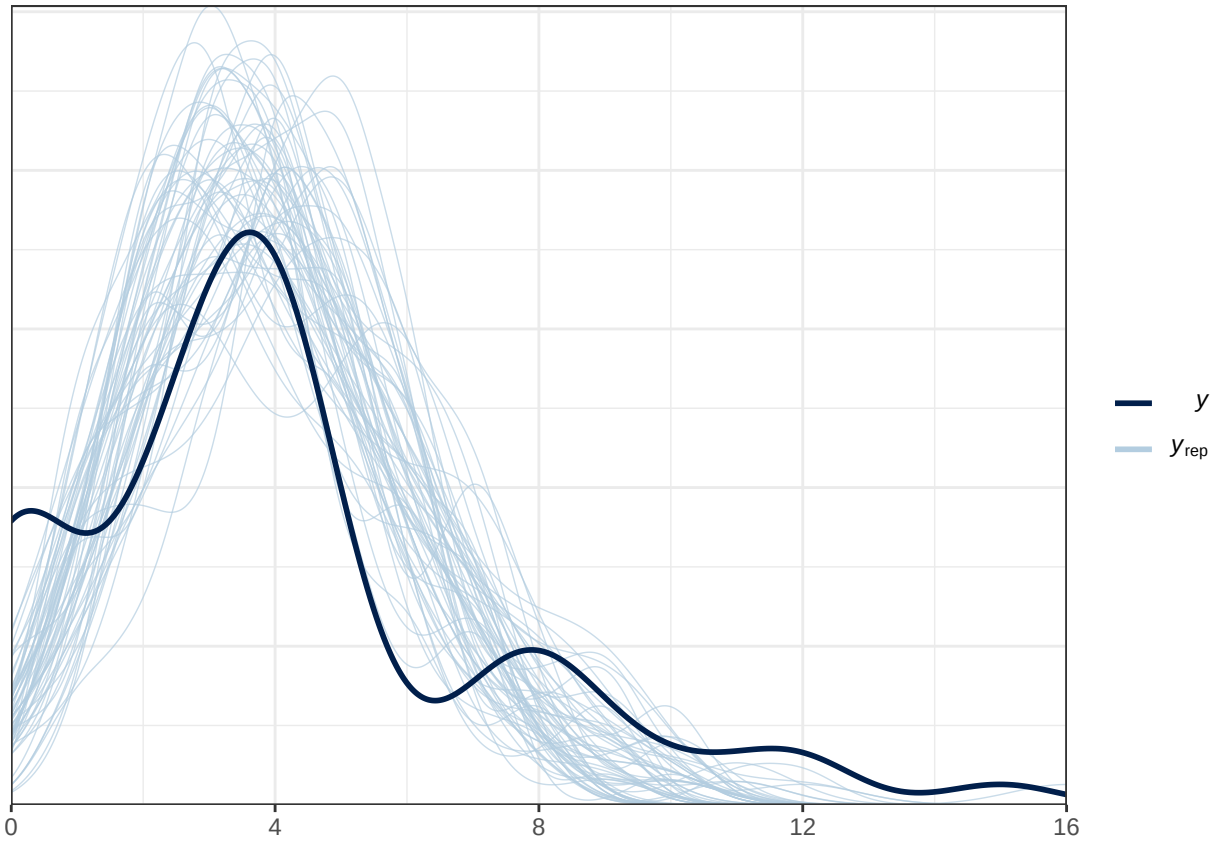


Exercise 4

Based on the PPC plots below, it looks like our model doesn't fit very well. We're not capturing the same proportion of 0s that we see in the sample. The variance of our sample is also much higher (if I'm reading

that plot correctly).

```
ppc_dens_overlay(y, yrep_matrix[1:60, ])
```

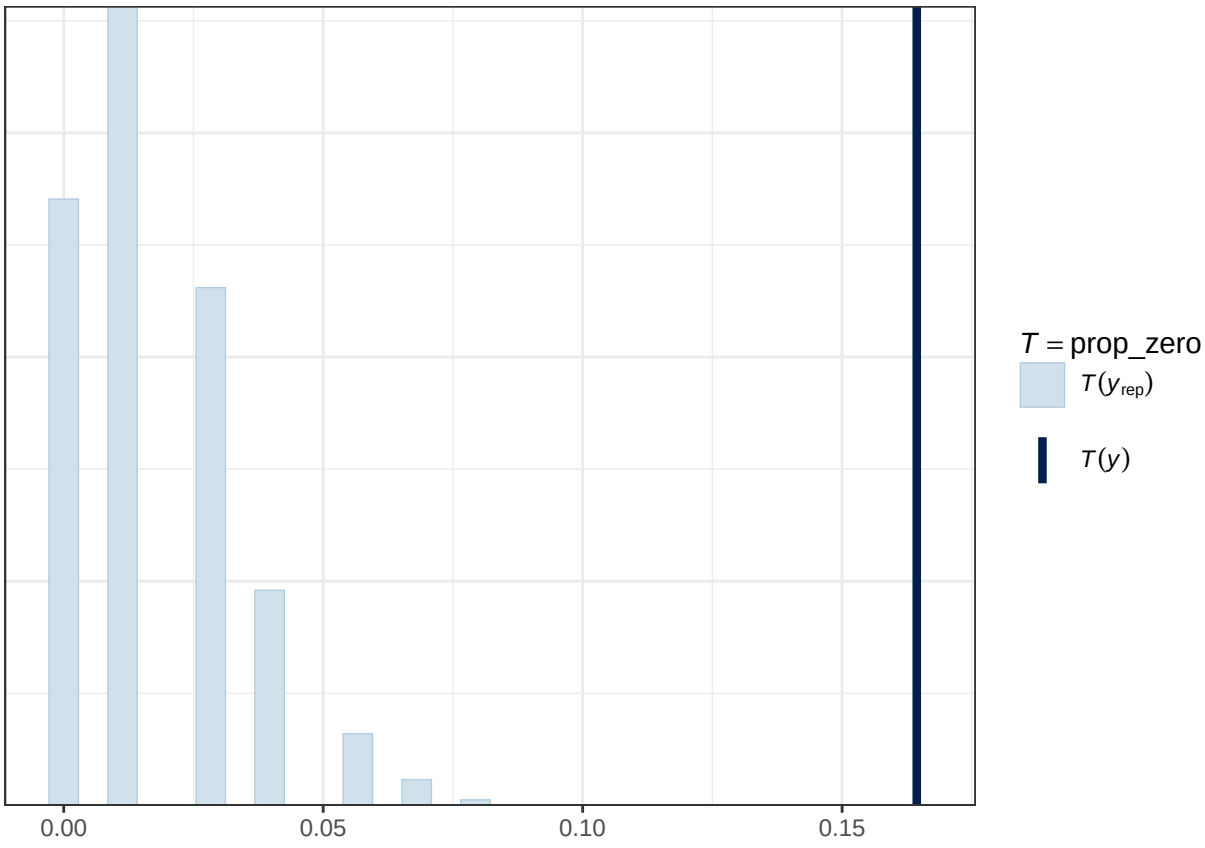



```
prop_zero <- function(x){
  mean(x == 0)
}
prop_zero(y)
```

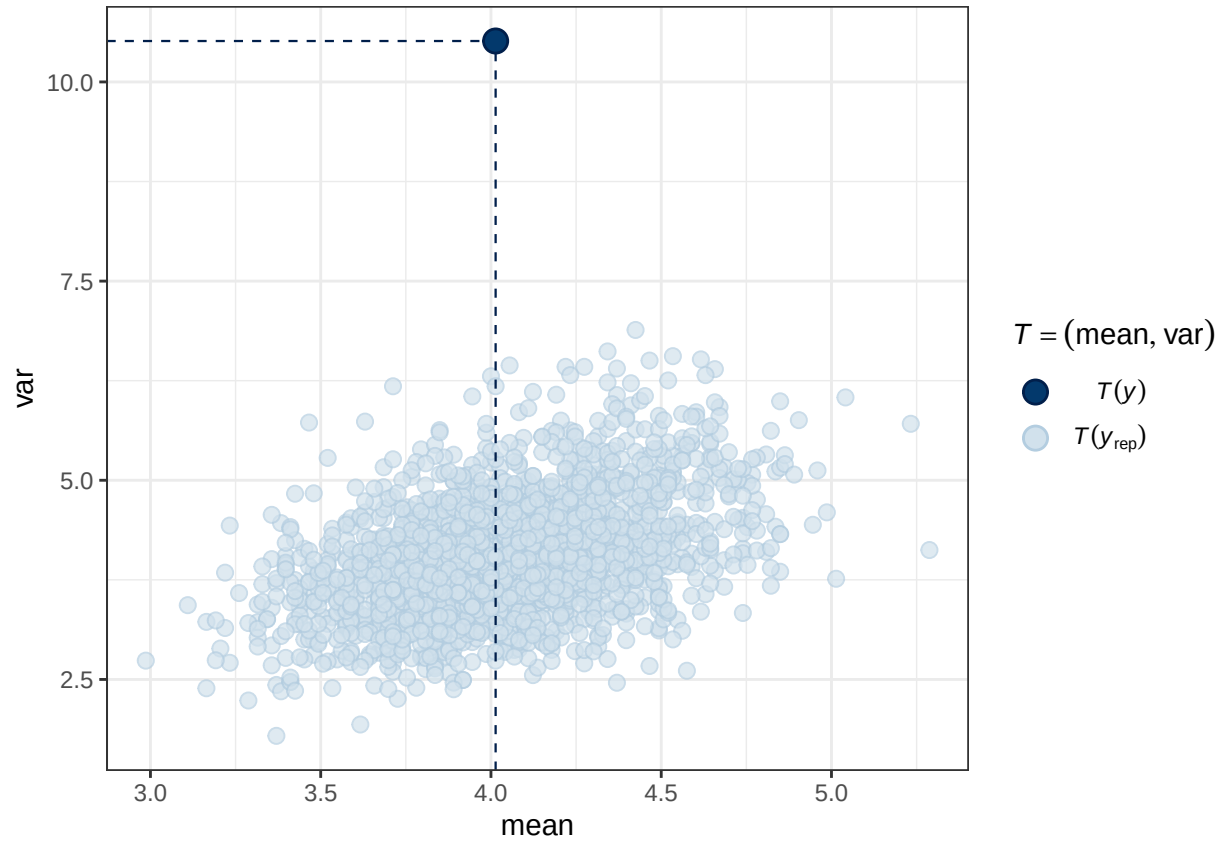
```
## [1] 0.1643836
```

```
ppc_stat(y, yrep_matrix, stat = "prop_zero")
```

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

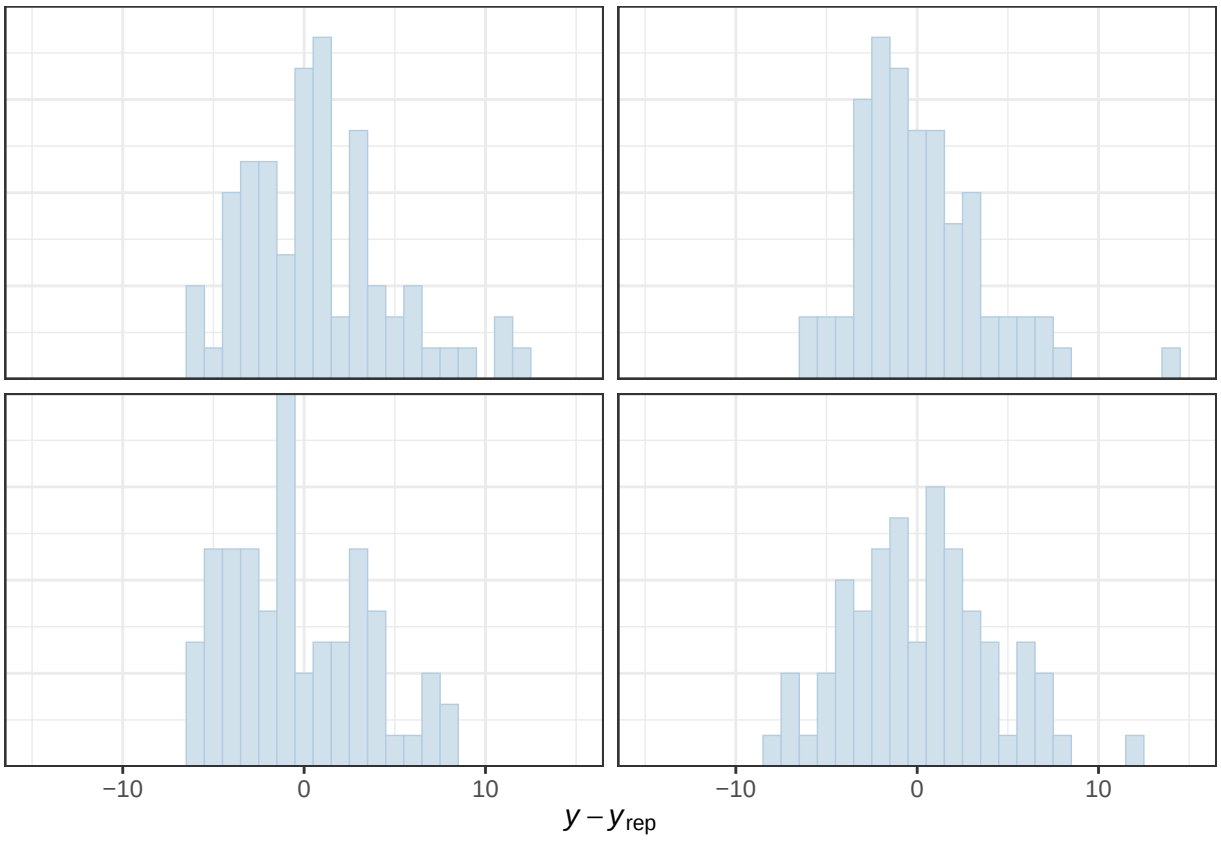



```
ppc_stat_2d(y, yrep_matrix, stat = c("mean", "var"))
```




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

```
## Warning: Removed 8 rows containing missing values (geom_bar).
```



Exercise 5

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

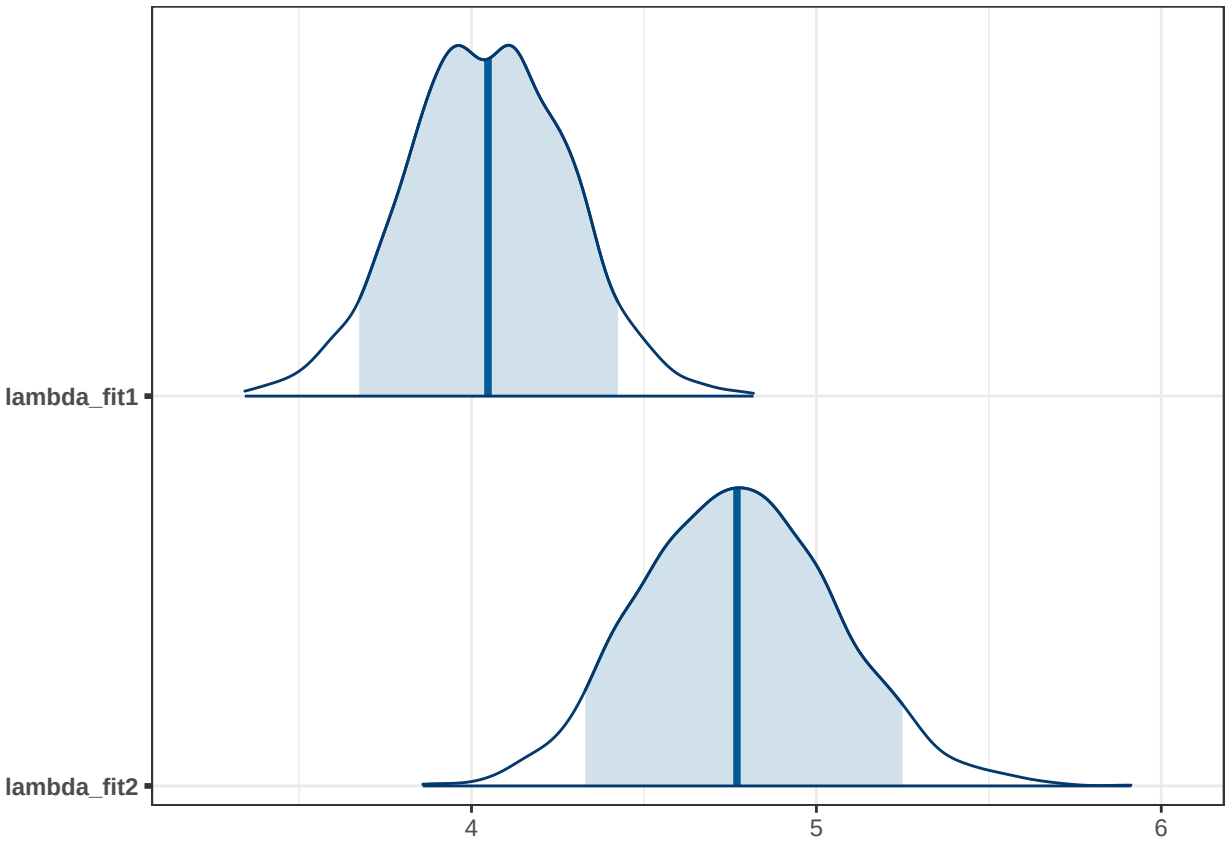
```
## Trying to compile a simple C file
```

```
## Running /usr/lib64/R/bin/R CMD SHLIB foo.c
## gcc -m64 -I"/usr/include/R" -DNDEBUG   -I"/usr/lib64/R/library/Rcpp/include/"  -I"/usr/lib64/R/libra
## In file included from /usr/lib64/R/library/RcppEigen/include/Eigen/Dense:1,
##                  from /usr/lib64/R/library/StanHeaders/include/stan/math/prim/mat/fun/Eigen.hpp:13,
##                  from <command-line>:
## /usr/lib64/R/library/RcppEigen/include/Eigen/Core:82:12: fatal error: new: No such file or directory
##    82 |   #include <new>
##       |            ^~~~~
## compilation terminated.
## make: *** [/usr/lib64/R/etc/Makeconf:168: foo.o] Error 1
```

```
# 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)
```

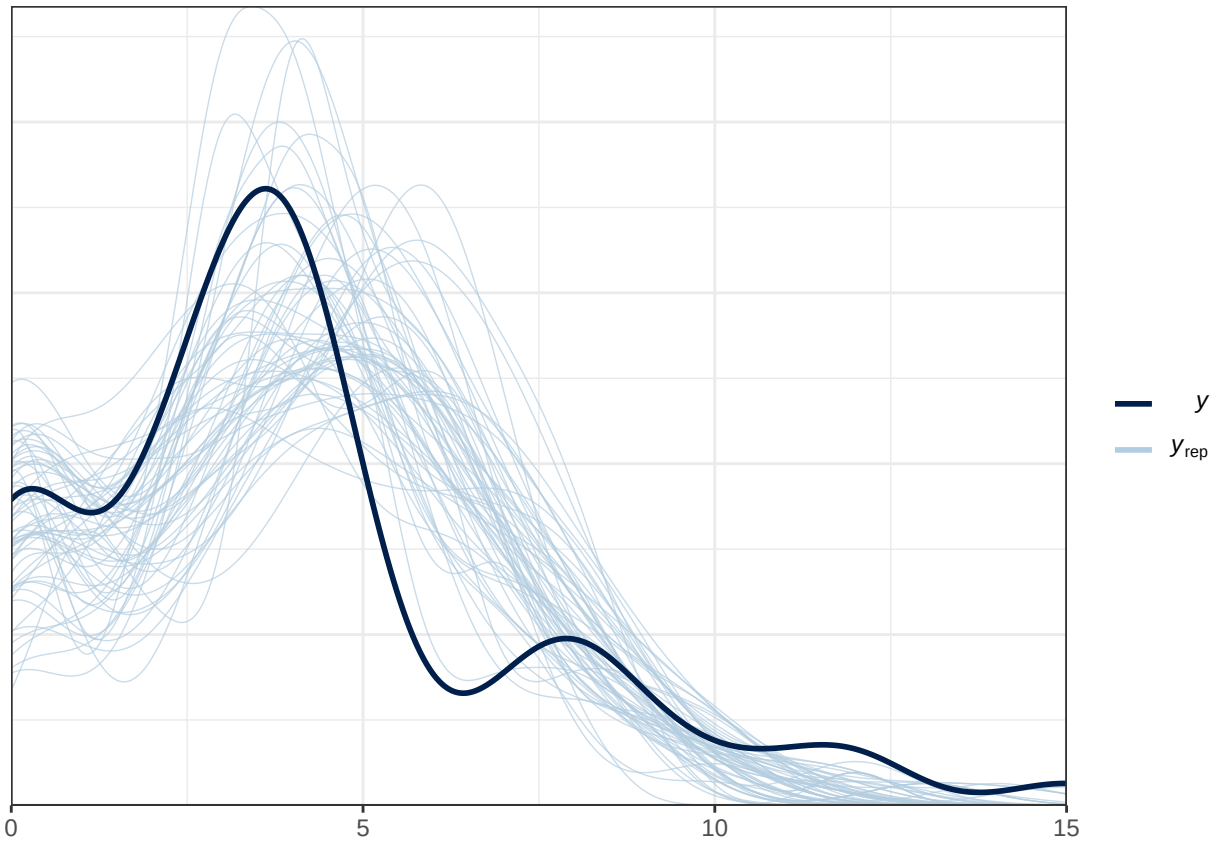


Exercise 6

From the plots, it looks like the second model performs much better and has a mean closer to the sample data.

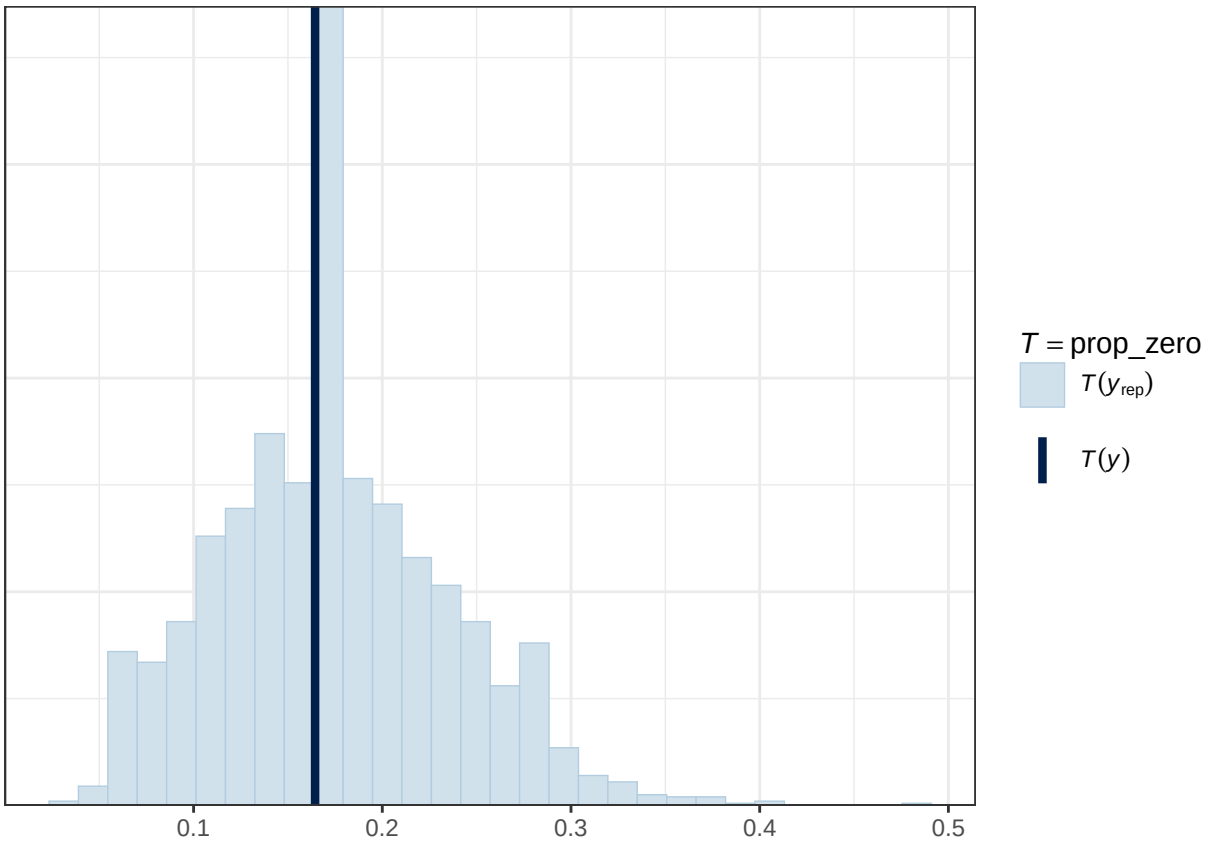
```
y_rep2 <- as.matrix(fit2, pars = "y_rep")

ppc_dens_overlay(y, y_rep2[1:60, ])
```

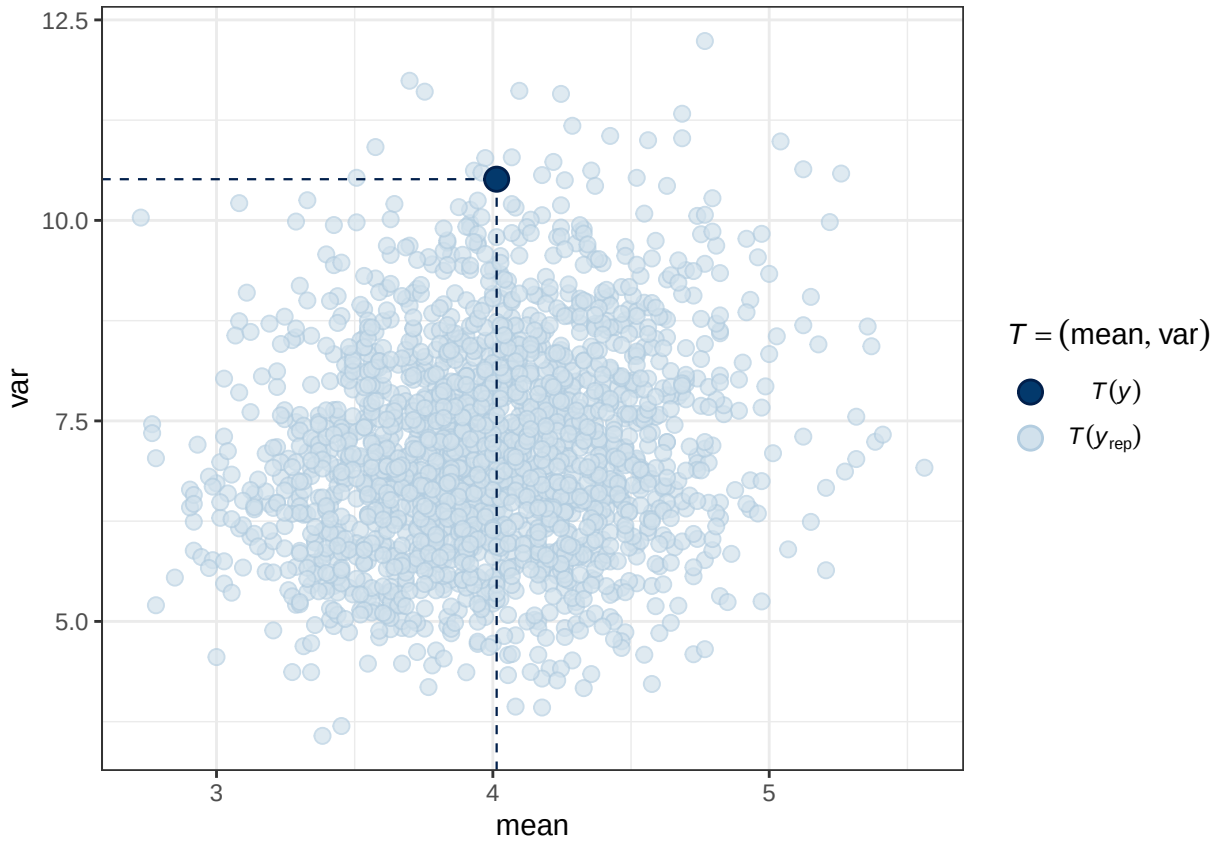



```
ppc_stat(y, y_rep2, stat = "prop_zero")
```

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

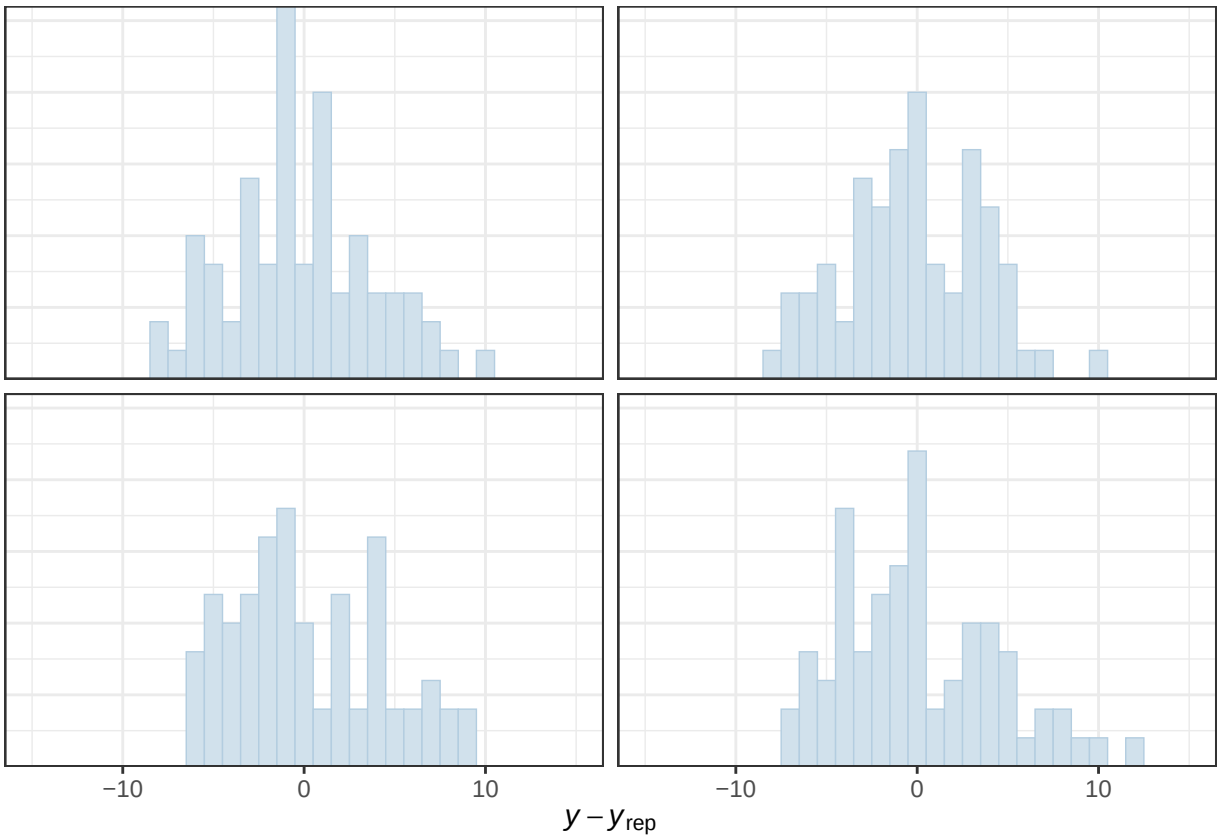



```
ppc_stat_2d(y, y_rep2, stat = c("mean", "var"))
```




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

```
## Warning: Removed 8 rows containing missing values (geom_bar).
```



Exercise 7

It looks like the second model has a lower SE so I'm going to guess that model performs a little better.

```
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   -203.2 14.5
## p_loo         2.5  0.5
## looic       406.3 29.0
## ------
## 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   -183.3 10.8
## p_loo         2.8  0.5
## looic       366.6 21.6
## ------
## 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.
```

Exercise 8

The importance of PPCs is to make sure our model is predicting accurately.

Exercise 9

The second model certainly looks like a better fit than the first. The proportions of zeros are closer to the sample, although the variance isn't quite as high as the sample. This is based on viewing the plots and the LOOCV (though I'm having a difficult time interpreting the LOOCV output, despite checking ?loo and the vignettes for it.)

Exercise 10

I don't think a single LOOCV would make sense - you would want at least two so you could compare the errors with each other. They make sense relative to each other.