Eliciting prior and posterior predictive distributions through MC approximation

CSSS/STAT 564: Bayesian Statistics for the Social Sciences

William Brown and Alex Jiang

CS&SS 564 (Spring 2021)

Goals

- Illustrate plug-in approximation of DAG-based joint probability distributions and the marginal distributions they imply
- A practical guide for eliciting prior predictive distributions and posterior predictive distributions
- Predicting identically generated data sets

Prediction in general

- Our intention in fitting probability models to data is often to understand the DGP underlying the data, often for the sake of making informed decisions about future outcomes.
- Suppose that the underlying data generating process is as follows: Y is generated according to a distribution characterized by (possibly unknown) parameter θ_m -- $Y \sim \mathrm{Dist}_m(\theta_m)$. Our goal is to describe the distribution of new values $\tilde{Y}$ coming from such distribution, before there are observed.
    - such procedure of making inferences about an unknown observable is often called **predictive inference**.
    - Let $\widetilde{Y}$ denote a "predictand"---a class of unknown quantities for which we intend to estimate future values. this quantity has the same domain as realized values of Y: $\widetilde{Y} = y \in \mathcal{Y}$. For greater succinctness, we can abbreviate the expression $\widetilde{Y} = y$ more simply as $\tilde{y}$.

Question: how do we do predictive inference, given the model we just described?

- If $Y \sim \mathrm{Dist}_m(\theta_m)$ and we knew θ_m exactly, then $\mathrm{Dist}_m(\theta_m)$ would double as a **predictive distribution**, i.e. $\widetilde{Y} \sim \mathrm{Dist}_m(\theta_m)$
    - However, we never know the true value θ_m (at least practically speaking). All we know (based on the model we specified) is the prior distribution of θ.

Bayesian prediction framework allows us to incorporate our (modeled) uncertainty regarding this fixed but unknown quantity.

- When we don't know θ_m, our prediction of $\widetilde{Y}$ requires us to elicit predictive probabilities $p(\tilde{y})$ by applying the Rule of Total Probability to the joint distribution $p(\tilde{y}, \theta_m)$:

$$p(\tilde{y}) = \int_{\Theta_m} p(\tilde{y}, \theta_m)\ d\theta_m$$

- The joint probability function $p(\tilde{y}, \theta_m)$ can be decomposed into (A) a model for the DGP underlying $\widetilde{Y}$ conditional on θ_m, and (B) a model for our uncertainty/belief regarding θ_m:

$$p(\tilde{y}, \theta_m) = p(\tilde{y}|\theta_m) \times p(\theta_m)$$

$$\therefore p(\tilde{y}) = \int_{\Theta_m} p(\tilde{y}|\theta_m)\ p(\theta_m)\ d\theta_m$$

- An interpretation: Ideally we would want to sample $\tilde{y}$ from $p(\tilde{y}|\theta_{true})$, where θ_{true} is the true value of the unknown parameter. Since we don't know the true value of θ_m, we incorporate our prior belief about this parameter to 'faciliate our guess', by taking a 'weighted average' wrt the prior distribution of θ_m.

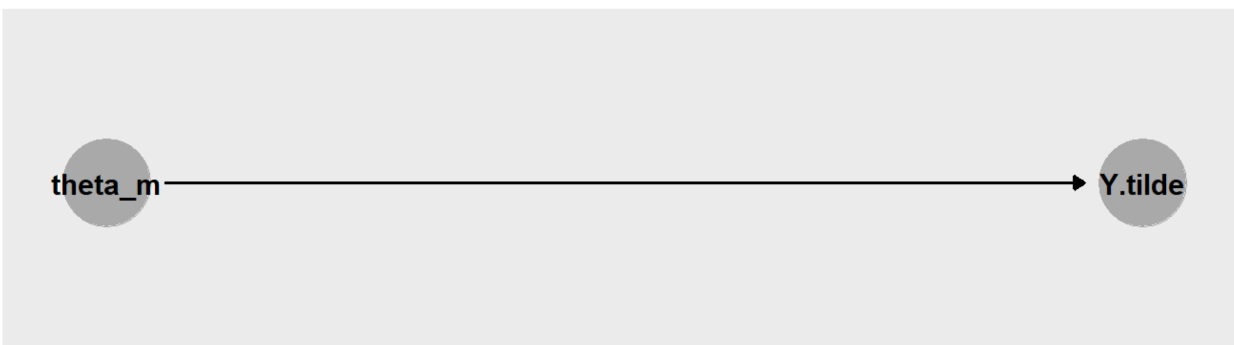
Depending on whether we have observed data $Y = y$ coming from the same process, the predictive distribution we use to describe the unknown data $\tilde{y}$ can be categorized into **prior predictive distribution** and **posterior predictive distribution**.

- the marginal distribution $p(\tilde{y})$ above is also called the **prior predictive distribution**, because:
    - (**prior**): it is not based on observed data:
    - (**predictive**): it is the distribution of $\tilde{y}$, which is observable (we're about to observe it) but unknown (we haven't observed it yet)

We will learn how to sample values from the prior predictive distribution through plug-in approximation in the following section:

- This implies a very simple DAG decomposition of the full probability model:

```
# Illustration of a DAG for a univariate DGP when the model parameters are unknown
theme_set(theme_dag_gray())

simpleDAG.coords = list(
  x = c(
    theta_m = 1, Y.tilde = 2
    ),
  y = c(
    theta_m = 1, Y.tilde = 1
    )
  )

simpleDAG = dagify(

  Y.tilde ~ theta_m, # Child on the left of the tilde, parent(s) on the right of the t
ilde separated by '+'

  coords = simpleDAG.coords
  )

simpleDAG %>%
  ggdag() +
  geom_dag_node(color = "dark gray", internal_color = "dark gray") +
  geom_dag_text(col = "black")
```

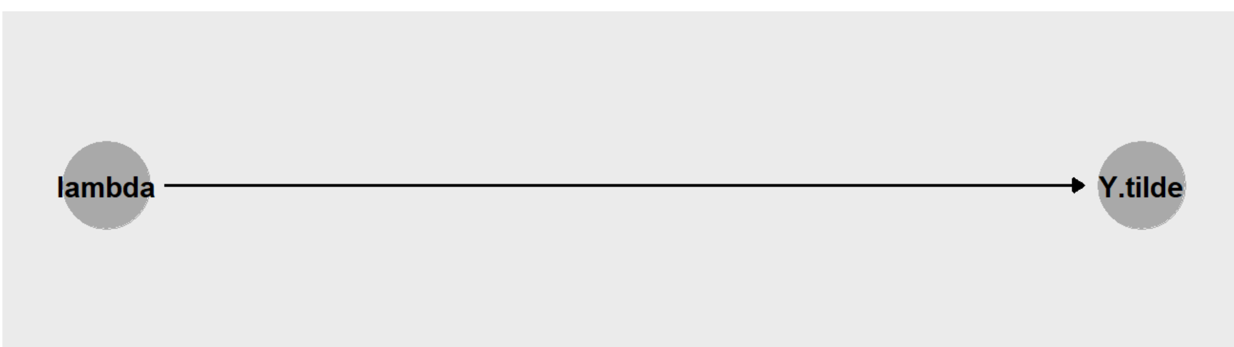


Plug-in approximation of prior predictive distributions---the case of the gamma-Poisson model

- While there are a small handful of analytical solutions for marginal predictive distributions $p(\tilde{y})$, these are mostly for "toy" problems. In most realistic data-analytic circumstances, plug-in approximation is favored as an efficient alternative.
- Basic procedure: draw a large MC sample from the joint PD of $\widetilde{Y}$ and θ_m, then summarize only the marginal sample distribution of $\widetilde{Y}$.
- Recall: sampling from a joint PD model based on a DAG is straightforward:
  - Simulate S observations from the DAG's root nodes (quantities represented by nodes with no parents): $\{\theta_m^{(1)}, \ldots, \theta_m^{(S)}\} \overset{iid}{\sim} \text{Dist}_{\theta_m}$

  - Pass this output as input for the simulation of child quantities: $y^{(s)} \sim \text{Dist}_m\left(\theta_m^{(s)}\right)$
  - Repeat this process until the terminal nodes are reached (quantities represented by nodes with no children).
- Where does $p(\theta_m)$ come from?
  - In general, it is some degree-of-belief budget about this uncertain quantity, $p(\theta_m|\eta)$, where η are the hyper-parameters of this model.
  - It could be a prior PD model, $p(\theta_m|\eta_0)$, where η_0 are the prior hyper-parameters.
  - It could be a posterior PD model, $p(\theta_m|\eta_0, y)$, where η_0 are again the prior hyper-parameters and y abbreviates observed data used to further condition or update our belief about θ_m.

```
# Illustration of a DAG for a univariate Poisson DGP when lambda is unknown
PriorPredDistDAG.coords = list(
  x = c(
    lambda = 1, Y.tilde = 2
    ),
  y = c(
    lambda = 1, Y.tilde = 1
    )
  )

PriorPredDistDAG = dagify(

  Y.tilde ~ lambda,

  coords = PriorPredDistDAG.coords
  )

PriorPredDistDAG %>%
  ggdag() +
  geom_dag_node(color = "dark gray", internal_color = "dark gray") +
  geom_dag_text(col = "black")
```



$$\tilde{y} \in \{0, 1, 2, \ldots\} \qquad \lambda \in \mathbb{R}_{>0}$$

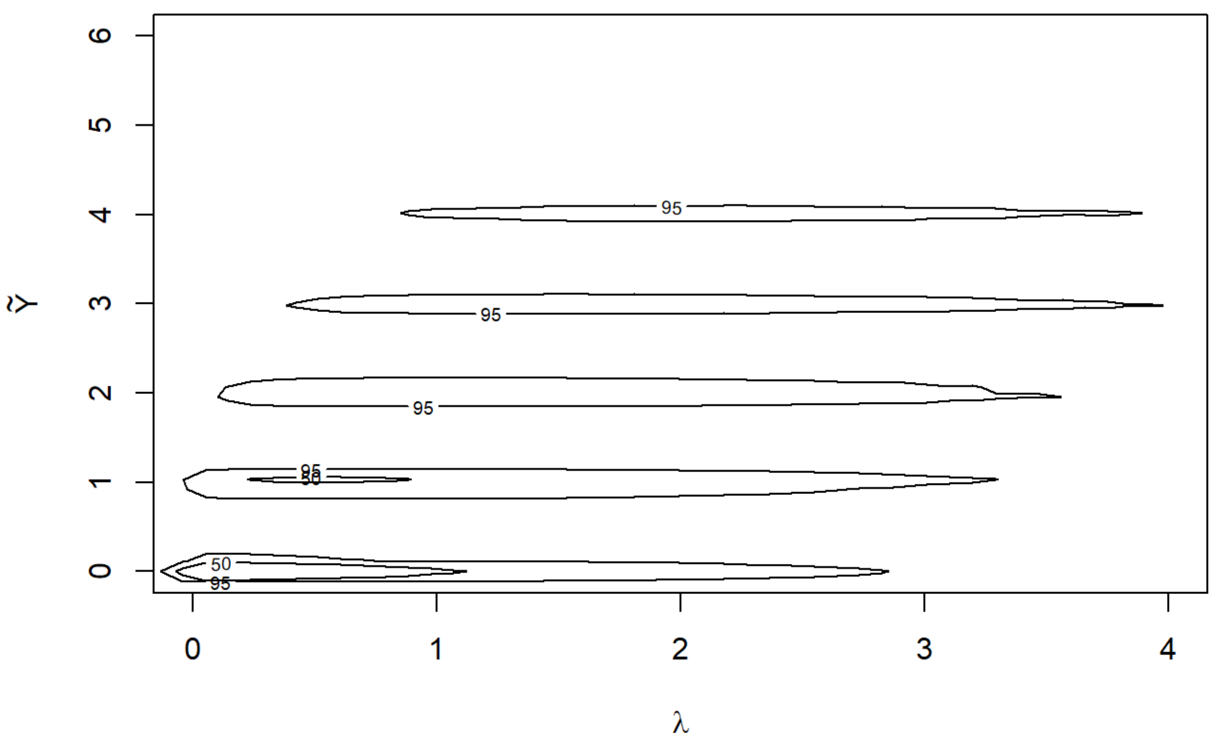
$$p(\tilde{y}, \lambda|a_0, b_0) = P(\tilde{y}|\lambda) \times p(\lambda|a_0, b_0)$$

$$\widetilde{Y} \sim \text{Poisson}(\lambda)$$

$$\lambda \sim \text{Gamma}(a_0, b_0)$$

a_0 is a prior shape hyper-parameter and b_0 is a prior concentration hyper-parameter.

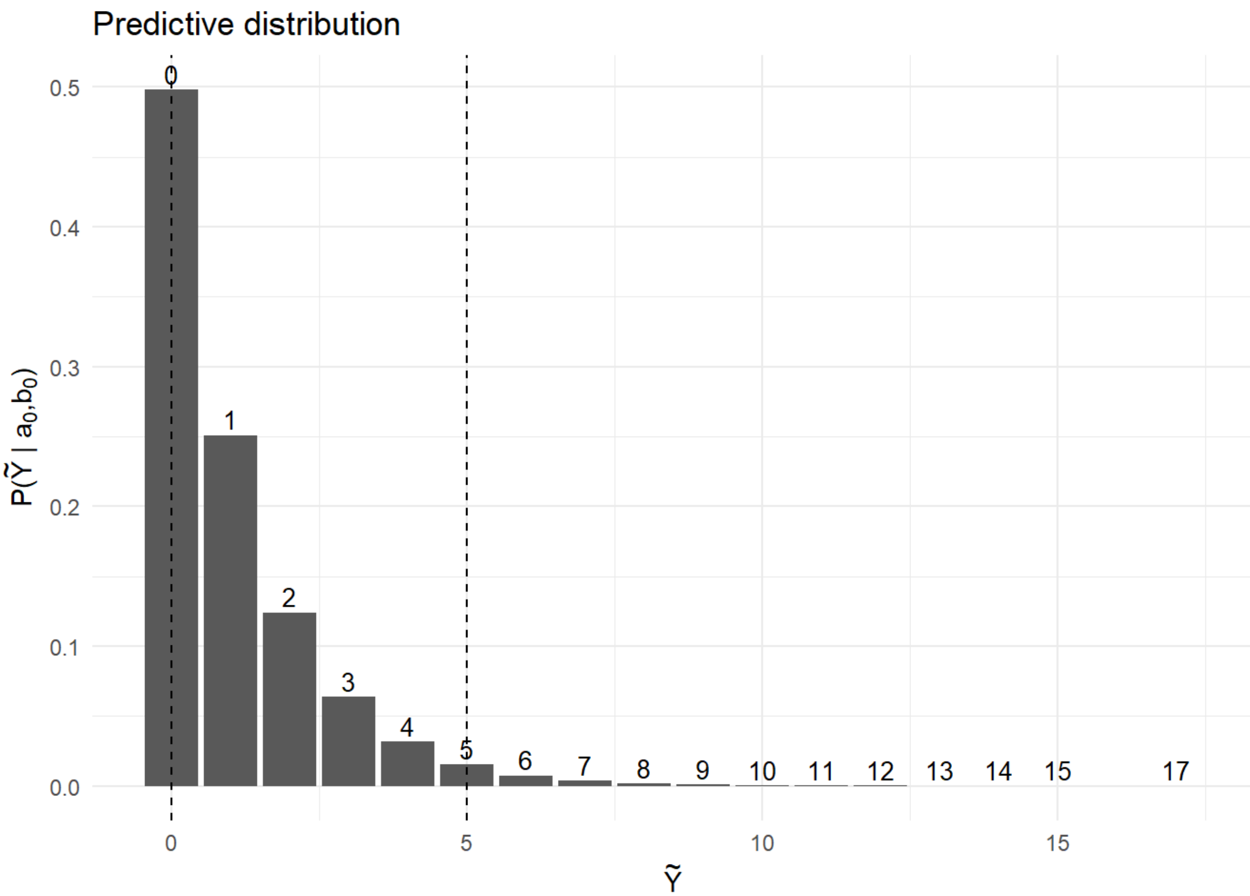
```
# MC simulation of the joint distribution of lambda and the Poisson-distributed estima
nd based on a prior model of lambda
a_0 = b_0 = 1; S=100000; set.seed(564)
lambdaSim = rgamma(n = S, shape = a_0, rate = b_0)
Y.tildeSim = rpois(n = S, lambda = lambdaSim)

# Illustration of the approximated joint prior distribution of lambda and the Poisson-
distributed estimand
jointPlugInApprox = ks::kde(x = cbind(lambdaSim, Y.tildeSim))
plot(
  x=jointPlugInApprox, # For help plotting kde objects, see ?plot.kde
  cont = c(50, 95), # Percentiles to draw highest-probability regions
  xlim = c(0,4), ylim = c(0, 6),
  xlab = expression(lambda), ylab = expression(tilde(Y))
  )
```



(Note that this contour plot treats $\widehat{Y}$ as if it were continuous rather than discrete.)

- The marginal prior predictive distribution of $\widetilde{Y}$ can then be approximated by plotting out only the simulated vector $\{\widetilde{Y}^{(1)}, \ldots, \widetilde{Y}^{(S)}\}$:

```
# Illustration of the approximated marginal prior distribution of the Poisson-distribu
ted estimand
df <- tibble(x = as.integer(rownames(table(Y.tildeSim))),
             y = table(Y.tildeSim)/S)
ggplot(df, aes(x=x, y=y)) + geom_bar(stat="identity") +
  geom_text(aes(label=x), vjust=-0.3, size=3.5)+theme_minimal() +
  xlab(expression(tilde(Y))) + ylab(expression(paste("P(",tilde(Y)," | a"[0],",b"[0],
")")))+
  geom_vline(xintercept = quantile(x = Y.tildeSim, probs = c(0.025,.975)), lty = 2) +
  ggtitle("Predictive distribution")
```



Plug-in prior predictive distribution approximation for the gamma-Poisson model. The dashed vertical line shows the 95% quantile-based credible interval.

Plug-in approximation of posterior predictive distributions---the case of the gamma-Poisson model

If we already observed data from the data-generating process y, the distribution of new unknown observable $\tilde{y}$, given the previous data $\tilde{y}$ (note their notational difference!) is called **posterior predictive distribution**, denoted as $p(\tilde{y}|y)$.

- How we derive $p(\tilde{y}|y)$ is very similar to what we did before: we first apply the Rule of Total Probability

$$p(\tilde{y} \mid y)=\int_{\Theta_m} p(\tilde{y}, \theta_m \mid y) d\theta_m$$

- We can similarly decompose the joint posterior probability function $p(\tilde{y}, \theta_m | y)$ into (A) a model for the DGP underlying $\tilde{Y}$ conditional on θ_m, **given** our prior knowledge of y, and (B) the posterior distribution for our unknown parameter θ_m given y:

$$p\left(\tilde{y}, \theta_m | y\right)=p\left(\tilde{y} \mid \theta_m, y\right) \times p\left(\theta_m | y\right)$$
$$\therefore p(\tilde{y} | y)=\int_{\Theta_m} p\left(\tilde{y} \mid \theta_m, y\right) p\left(\theta_m | y\right) d\theta_m$$

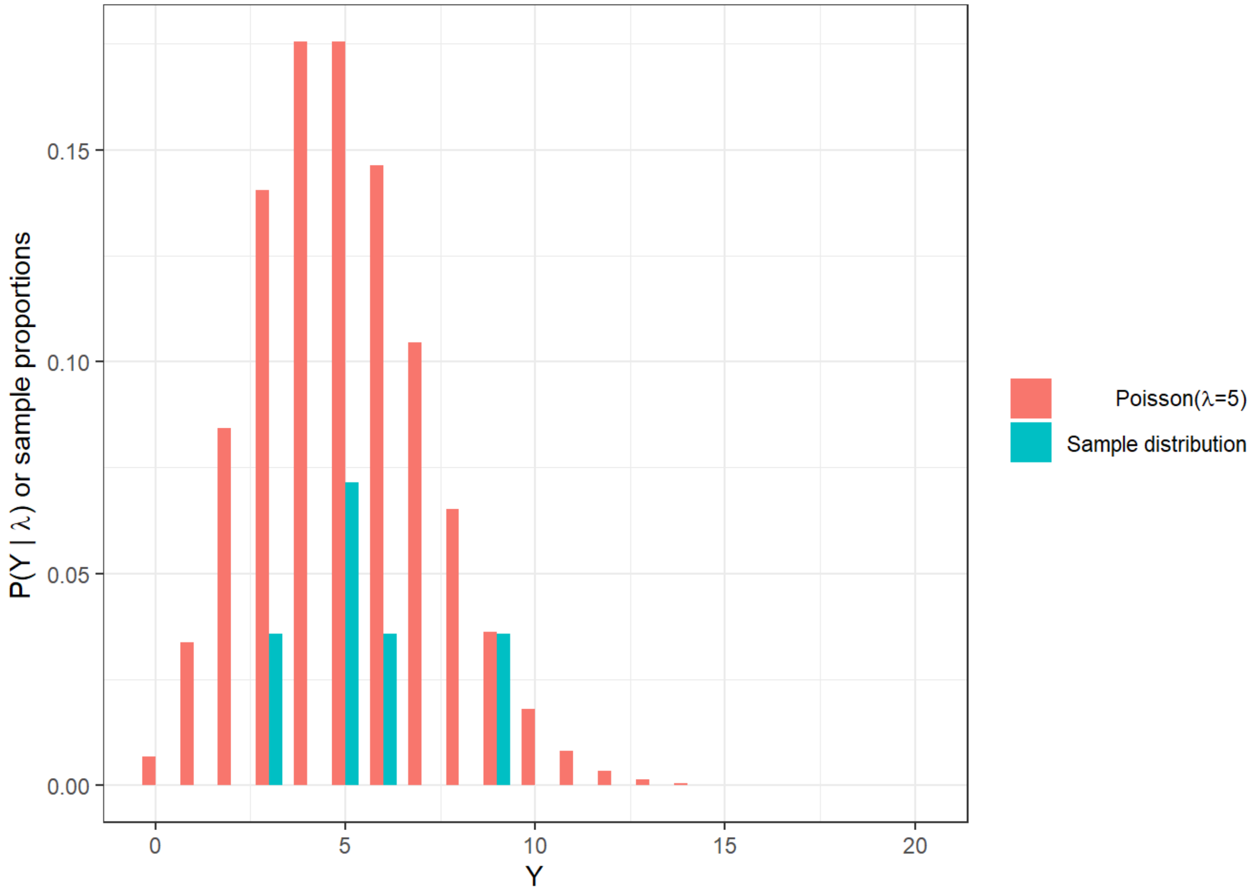
- Finally, based on our DAG model, we have **assumed conditional independence** of y and $\tilde{y}$ given θ:

$$p(\tilde{y} \mid y)=\int_{\Theta_m} p\left(\tilde{y} \mid \theta_m, y\right) p\left(\theta_m \mid y\right) d\theta_m=\int_{\Theta_m} p\left(\tilde{y} \mid \theta_m\right) p\left(\theta_m \mid y\right) d\theta_m$$

- Derivation of this result requires theory on d-separation for DAGs and is beyond the scope of today's lecture, but here's one possible interpretation for this: θ_m is the common and only cause for $\tilde{y}$ and y, so we wouldn't gain extra information about $\tilde{y}$ if we already controlled/conditioned on θ_m, even if we observed y in the first place.
- This is called 'Principle of the Common Cause' and you can find more information here if you're interested: https://www.andrew.cmu.edu/user/scheines/tutor/d-sep.html (https://www.andrew.cmu.edu/user/scheines/tutor/d-sep.html)
- To illustrate, let us first generate a sample of 5 observations from a known Poisson distribution:

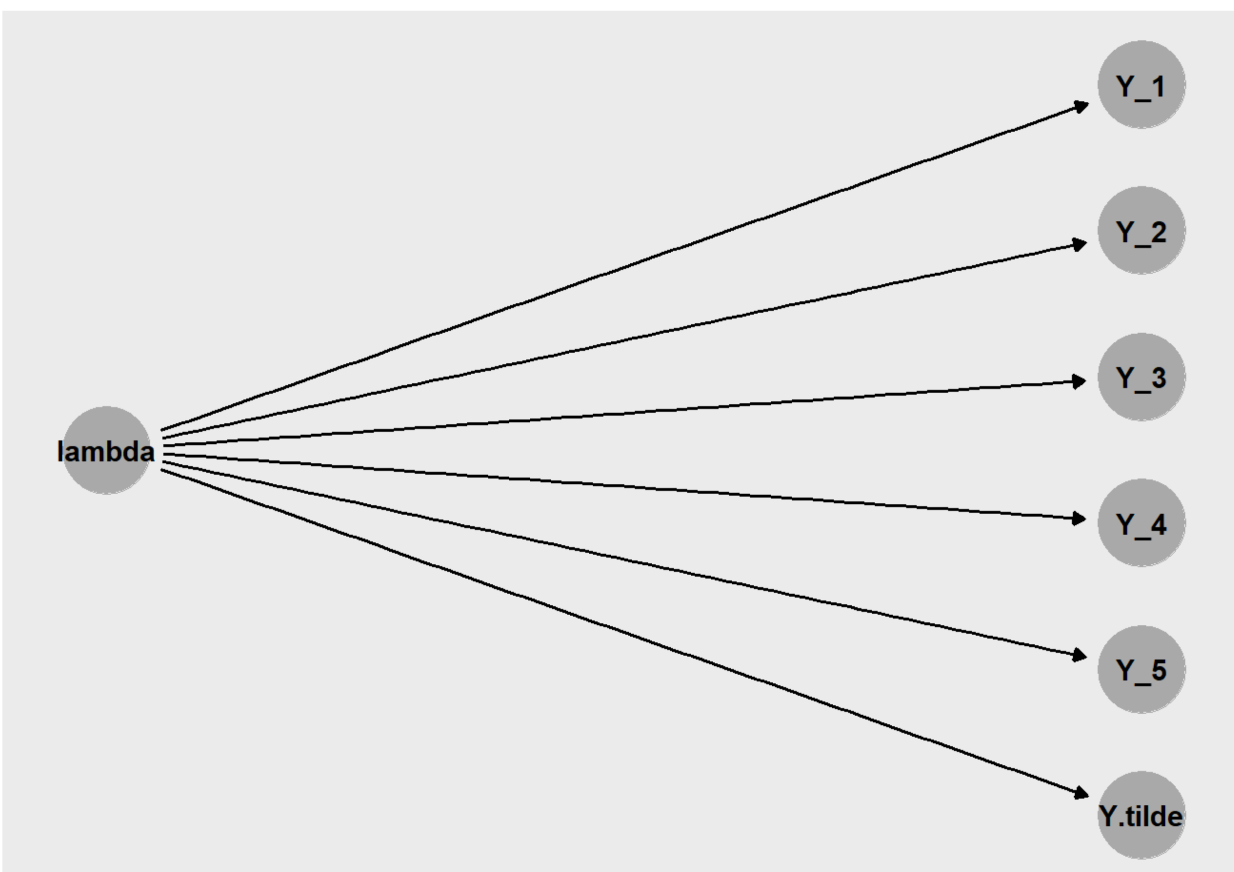
$$\{Y_1, \ldots, Y_5\} \stackrel{iid}{\sim} \operatorname{Poisson}(\lambda=5)$$

```
# Illustration of a known {Y_1,...,Y_5} ~ Poisson(lambda=5) sample
check_occurences <- function(x) {
  return(sum(Y.Sim %in% x))
}
Y.Sim = rpois(n = 5, lambda = 5)
df <- tibble(x = 0:20, y1 = dpois(x = 0:20, lambda = 5),
             y2 = sapply(0:20, check_occurences)/sum(Y.Sim)) %>%
  pivot_longer(!x,names_to = "type", values_to = "probs")
ggplot(df) + geom_bar(stat='identity',
                      aes(x = x, y = probs, fill = factor(type)),
                      position=position_dodge(width = .7), width = .7)+
  theme_bw() + xlab("Y") + ylab(expression(paste("P(Y | ", lambda,") or sample proport
ions"))) +
  theme(legend.title = element_blank()) +
  scale_fill_discrete(labels = c(expression(paste("Poisson(",lambda,"=5)"), "Sample di
stribution")))
```



- Eliciting posterior predictive distributions involves simulating from a joint distribution that includes not only the predictand and model parameter(s) but also observed data:

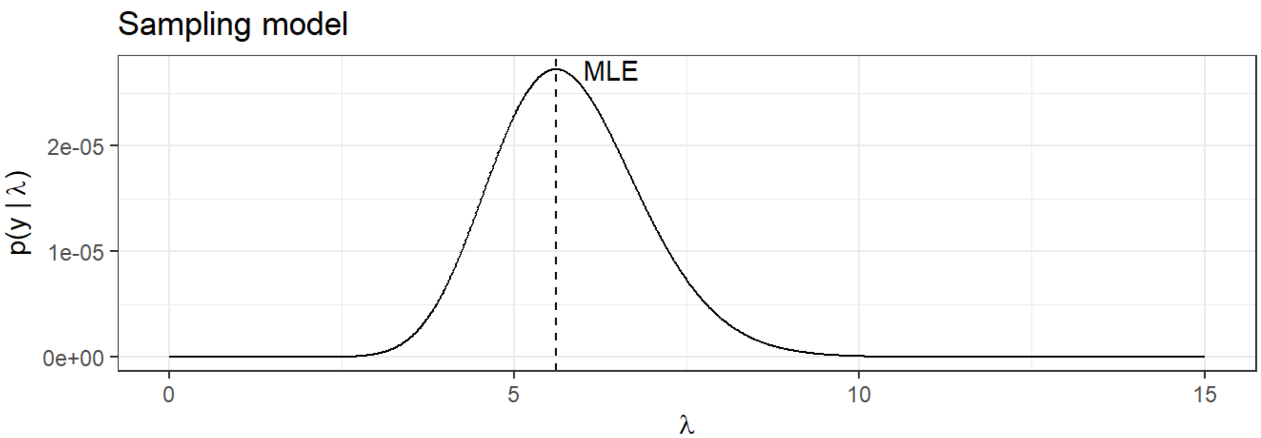
```
# Illustration of a DAG for a univariate Poisson DGP when lambda is unknown, with both
observed and unobserved/predicted values of Y
PostPredDistDAG.coords = list(
  x = c(
    lambda = 1, Y_1 = 2, Y_2 = 2, Y_3 = 2, Y_4 = 2, Y_5 = 2, Y.tilde = 2
    ),
  y = c(
    lambda = 1, Y_1 = 3.5, Y_2 = 2.5, Y_3 = 1.5, Y_4 = 0.5, Y_5 = -0.5, Y.tilde = -1.5
    )
  )

PostPredDistDAG = dagify(

  Y_1 ~ lambda,
  Y_2 ~ lambda,
  Y_3 ~ lambda,
  Y_4 ~ lambda,
  Y_5 ~ lambda,
  Y.tilde ~ lambda,

  coords = PostPredDistDAG.coords
  )

PostPredDistDAG %>%
  ggdag() +
  geom_dag_node(color = "dark gray", internal_color = "dark gray") +
  geom_dag_text(col = "black")
```

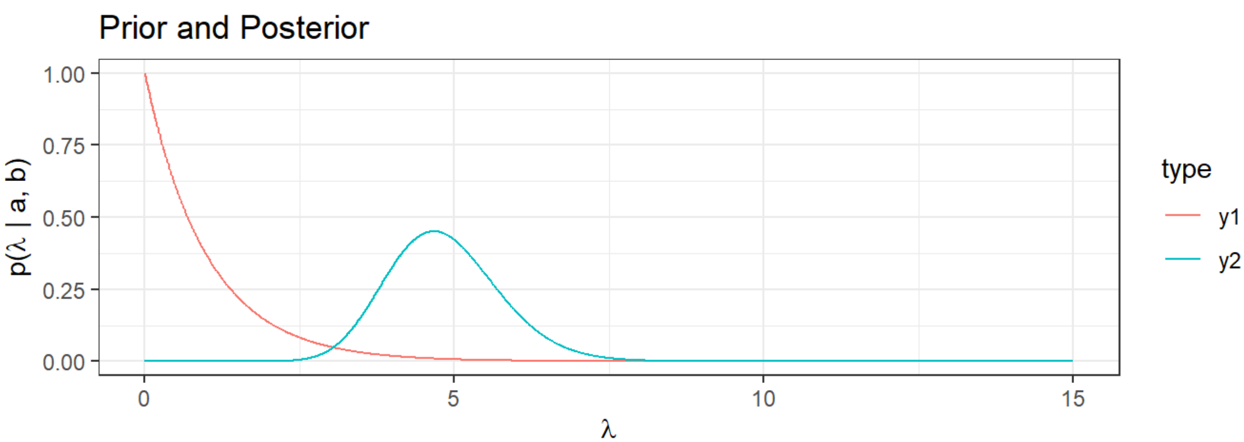


- However, in this case, we know several values represented in the joint distribution so we must apply a different strategy for sampling from the DAG. The easiest and fully sufficient way is to simulate from the posterior distribution of λ, then to simulate the predictand as before.
- For the gamma-Poisson model, an analytical solution is available for the posterior distribution with the following posterior shape and concentration hyper-parameters:

$$\lambda|y \sim \text{Gamma}\left(a_n = a_0 + \sum_{i=1}^{n} y_i, b_n = b_0 + n\right)$$

```
# MC simulation of the joint distribution of lambda and the Poisson-distributed estima
nd based on a posterior model of lambda
a_n = a_0+sum(Y.Sim); b_n = b_0+5
lambdaPostSim = rgamma(n = S, shape = a_n, rate = b_n)
Y.tildePostSim = rpois(n = S, lambda = lambdaPostSim)

# Illustration of the likelihood, prior, and posterior over Lambda
likelihoods = exp(colSums(apply(
    X = matrix(seq(0, 15, 0.01), ncol = 1),
    MARGIN = 1,
    FUN = dpois, x = Y.Sim, log = TRUE
    )))

df1 <- tibble(x = seq(0, 15, 0.01), y = likelihoods)
p1 <- ggplot(df1, aes(x = x, y = y)) + geom_line() + xlab(expression(lambda)) +
  ylab(expression(paste("p(y | ",lambda,")"))) +
  geom_vline(xintercept = mean(Y.Sim), lty = 2) + theme_bw() +
  ggtitle("Sampling model")
p1 <- p1 + annotate("text", x = mean(Y.Sim)+0.8, y=max(likelihoods), label = "MLE")
df2 <- tibble(x = seq(0, 15, 0.01), y1 = dgamma(x=seq(0, 15, 0.01),
                                              shape = a_0, rate = b_0),
              y2 = dgamma(x=seq(0, 15, 0.01), shape = a_n, rate = b_n)) %>%
  pivot_longer(!x, names_to = "type", values_to = "prob")
p2 <- ggplot(df2, aes(x = x, y = prob, col = type, group = type)) + geom_line() +
  theme(legend.title = element_blank()) +
  ggtitle("Prior and Posterior") + theme_bw() +
  scale_fill_discrete(labels = c(
    expression(paste("Prior for ",lambda)),
    expression(paste("Posterior for ",lambda))
    )) + xlab(expression(lambda)) + ylab(expression(paste("p(",lambda," | a, b)")))
p <- grid.arrange(p1, p2, ncol = 1)
```

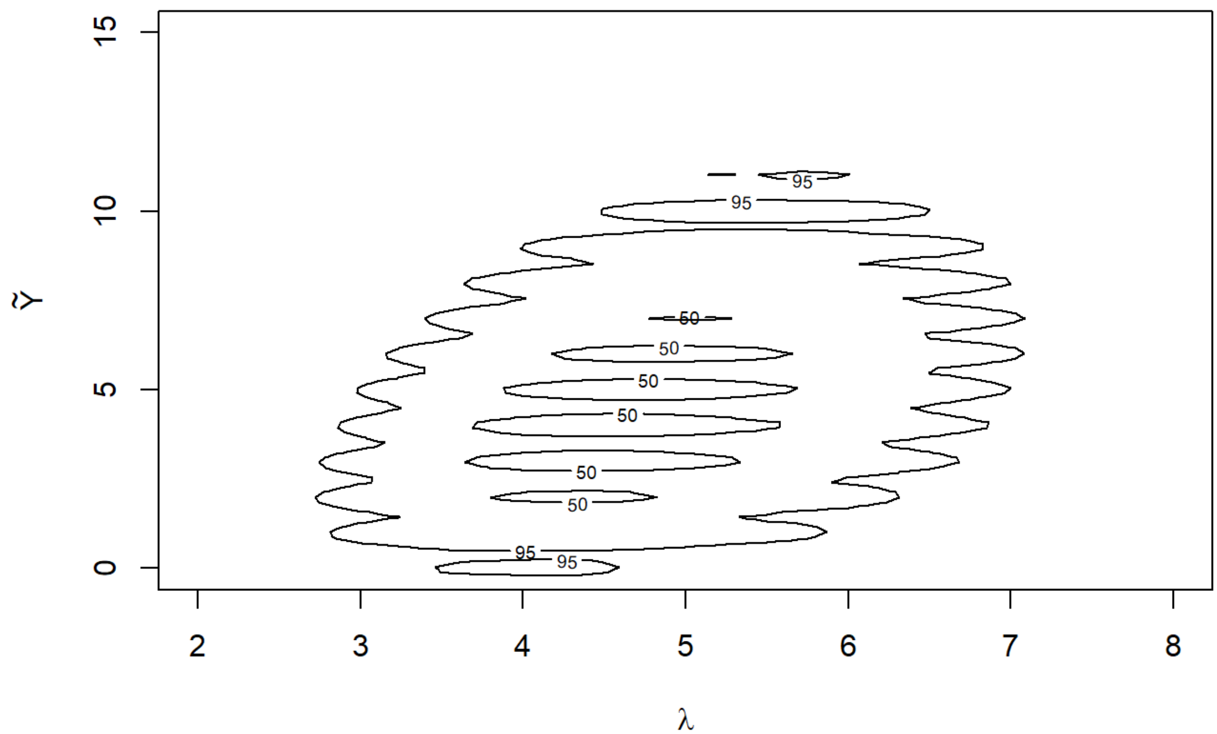





```
p
```

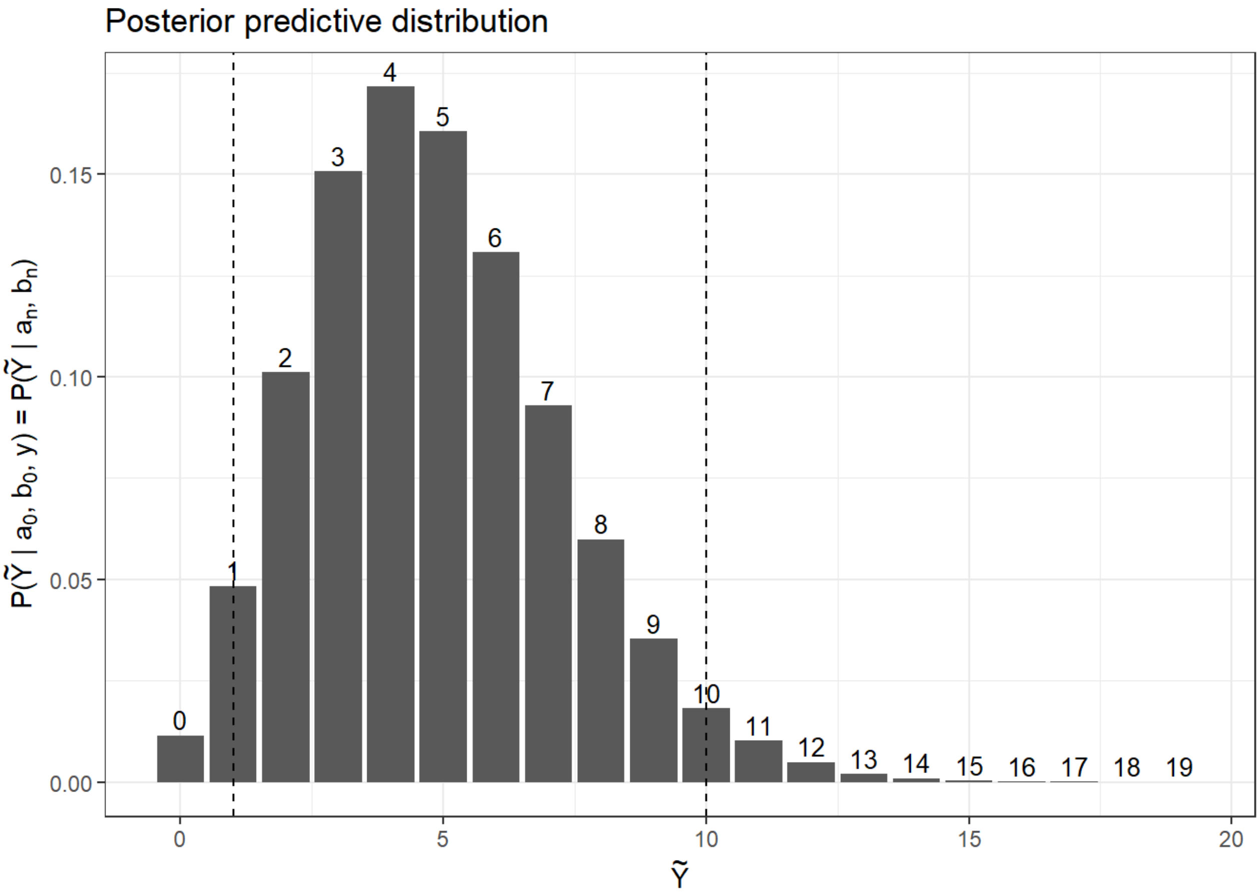
```
## TableGrob (2 x 1) "arrange": 2 grobs
##   z     cells    name           grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (2-2,1-1) arrange gtable[layout]
```

```
# Illustration of the approximated joint posterior distribution of lambda and the Pois
son-distributed predictand
jointPostPlugInApprox = ks::kde(x = cbind(lambdaPostSim, Y.tildePostSim))
par(mfrow = c(1, 1))
plot(
  x=jointPostPlugInApprox, # For help plotting kde objects, see ?plot.kde
  cont = c(50, 95), # Percentiles to draw highest-probability regions
  xlim = c(2,8), ylim = c(0, 15),
  xlab = expression(lambda), ylab = expression(tilde(Y))
  )
```



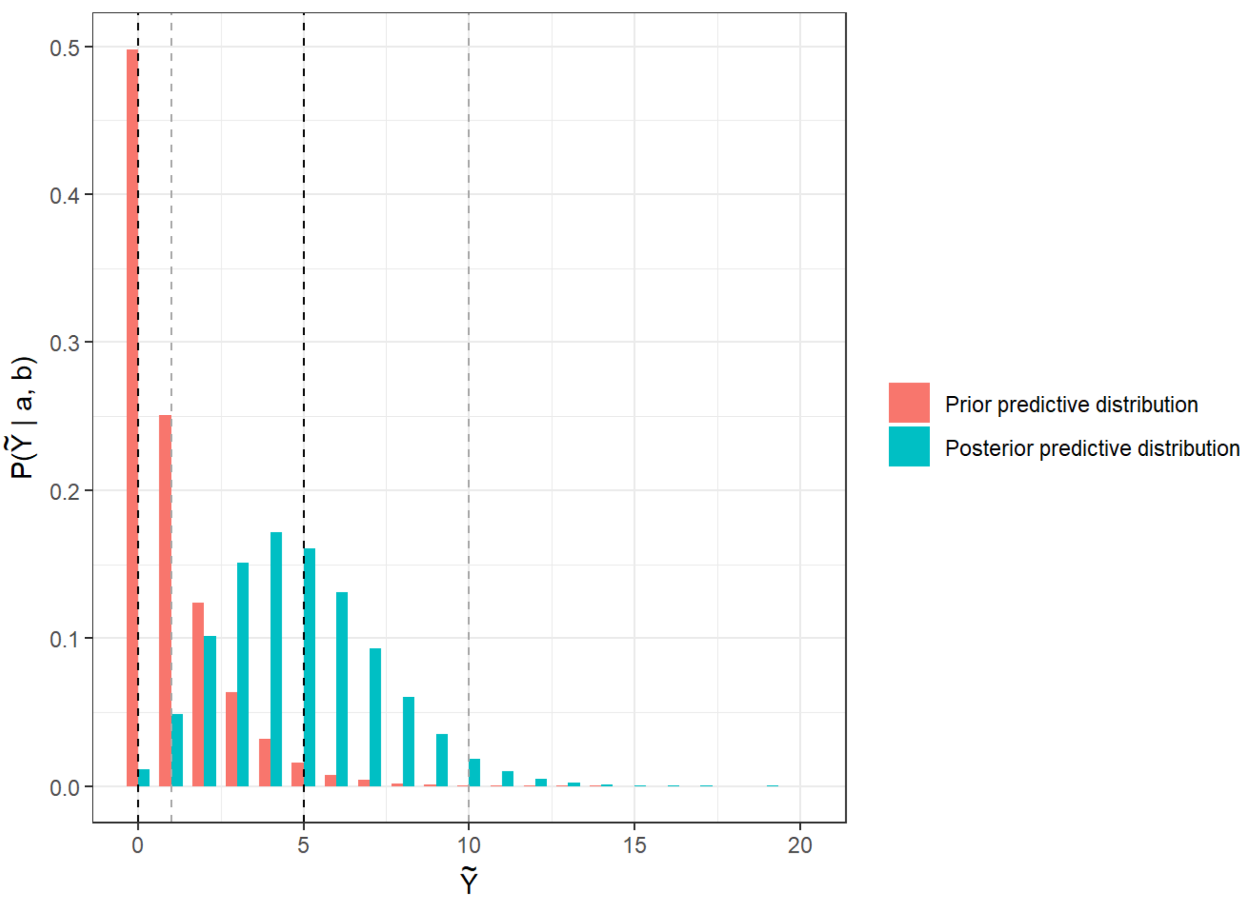

```
# Illustration of the approximated marginal posterior distribution of the Poisson-dist
ributed estimand
# 95% highest-probability predictive predictive interval
df <- tibble(x = as.integer(rownames(table(Y.tildePostSim))),
             y = table(Y.tildePostSim)/S)
ggplot(df, aes(x=x, y=y)) + geom_bar(stat="identity") +
  geom_text(aes(label=x), vjust=-0.3, size=3.5)+theme_minimal() +
  xlab(expression(tilde(Y))) + ylab(expression(paste("P(",tilde(Y)," | a"[0],", b"[0],
", y) = P(",tilde(Y)," | a"[n],", b"[n],")")))+
  geom_vline(xintercept = quantile(x = Y.tildePostSim, probs = c(0.025,.975)), lty = 2
) +
  ggtitle("Posterior predictive distribution")+ theme_bw()
```

```
## Don't know how to automatically pick scale for object of type table. Defaulting to
continuous.
```



Comparing prior and posterior predictive distributions

```
# Illustration comparing the approximated marginal prior and posterior distributions o
f the Poisson-distributed predictand
check_occurences_1 <- function(x) {
  return(sum(Y.tildeSim %in% x))
}
check_occurences_2 <- function(x) {
  return(sum(Y.tildePostSim %in% x))
}
df <- tibble(x = 0:20, y1 = sapply(0:20, check_occurences_1)/S,
             y2 = sapply(0:20, check_occurences_2)/S) %>%
  pivot_longer(!x, names_to = "type", values_to = "probs")
ggplot(df) + geom_bar(stat='identity',
                      aes(x = x, y = probs, fill = factor(type)),
                      position=position_dodge(width = .7), width = .7)+
  theme_bw() + xlab(expression(tilde(Y))) +
  ylab(expression(paste("P(",tilde(Y)," | a, b)"))) +
  geom_vline(xintercept = quantile(x = Y.tildeSim, probs = c(0.025,0.975)),
             lty = 2, col = "black") +
  geom_vline(xintercept = quantile(x = Y.tildePostSim, probs = c(0.025,0.975)),
             lty = 2, col = "dark gray") +
  theme(legend.title = element_blank()) +
  scale_fill_discrete(labels = c("Prior predictive distribution",
                                 "Posterior predictive distribution"))
```



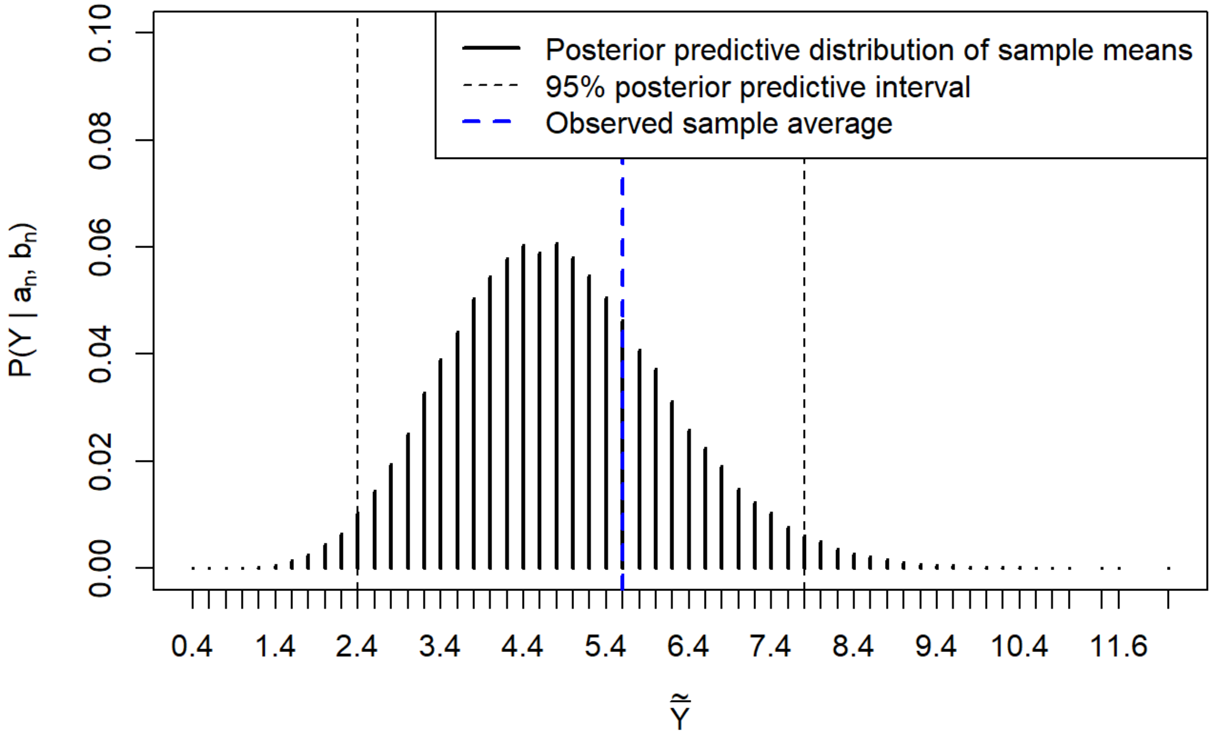
Simulating posterior (or prior) predictive distributions for whole samples

- Rather than drawing a single observation from the DGP model for each simulated value of θ_m, we can draw whole samples if we wish:

$$\{\widetilde{Y}_1^{(1)}, \ldots, \widetilde{Y}_n^{(1)}\} \overset{iid}{\sim} \text{Dist}_m(\theta_m^{(1)})$$
$$\{\widetilde{Y}_1^{(2)}, \ldots, \widetilde{Y}_n^{(2)}\} \overset{iid}{\sim} \text{Dist}_m(\theta_m^{(2)})$$
$$\vdots$$
$$\{\widetilde{Y}_1^{(S)}, \ldots, \widetilde{Y}_n^{(S)}\} \overset{iid}{\sim} \text{Dist}_m(\theta_m^{(S)})$$

- If (A) a particular summary statistic is calculated for each sample and (B) each sample is identical in size to the original sample, this simulation yields an approximation of a posterior predictive distribution of the chosen summary statistic, providing a Bayesian equivalent to a frequentist sampling distribution.

```
# Simulating a posterior predictive distribution of sample means for a Poisson model
Sample.tildePostSim = apply(
  X = as.matrix(lambdaPostSim, ncol=1), MARGIN = 1, FUN = rpois, n=5
  )
# Plotting the predictive distribution and comparing to the observed sample mean
plot(
  x=table(colMeans(Sample.tildePostSim))/S,
  xlab = expression(tilde(bar(Y))),
  ylab = expression(paste("P(",tilde(bar(Y)),"  | a"[n],", b"[n],")")),
  ylim = c(0, 0.1)
  )
abline(v = quantile(colMeans(Sample.tildePostSim), probs = c(0.025, 0.975)), lty = 2)
abline(v = mean(Y.Sim), lty = 2, col = "blue", lwd = 2)
legend(
  "topright",
  legend = c(
    "Posterior predictive distribution of sample means",
    "95% posterior predictive interval",
    "Observed sample average"
    ),
  lwd = c(2, 1, 2), lty = c(1, 2, 2), col = c("black", "black", "blue")
  )
```


Posterior predictive distribution of sample means
95% posterior predictive interval
Observed sample average
P(Y | a_n, b_n)
0.00
0.02
0.04
0.06
0.08
0.10
0.4
1.4
2.4
3.4
4.4
5.4
6.4
7.4
8.4
9.4
10.4
11.6
Ỹ