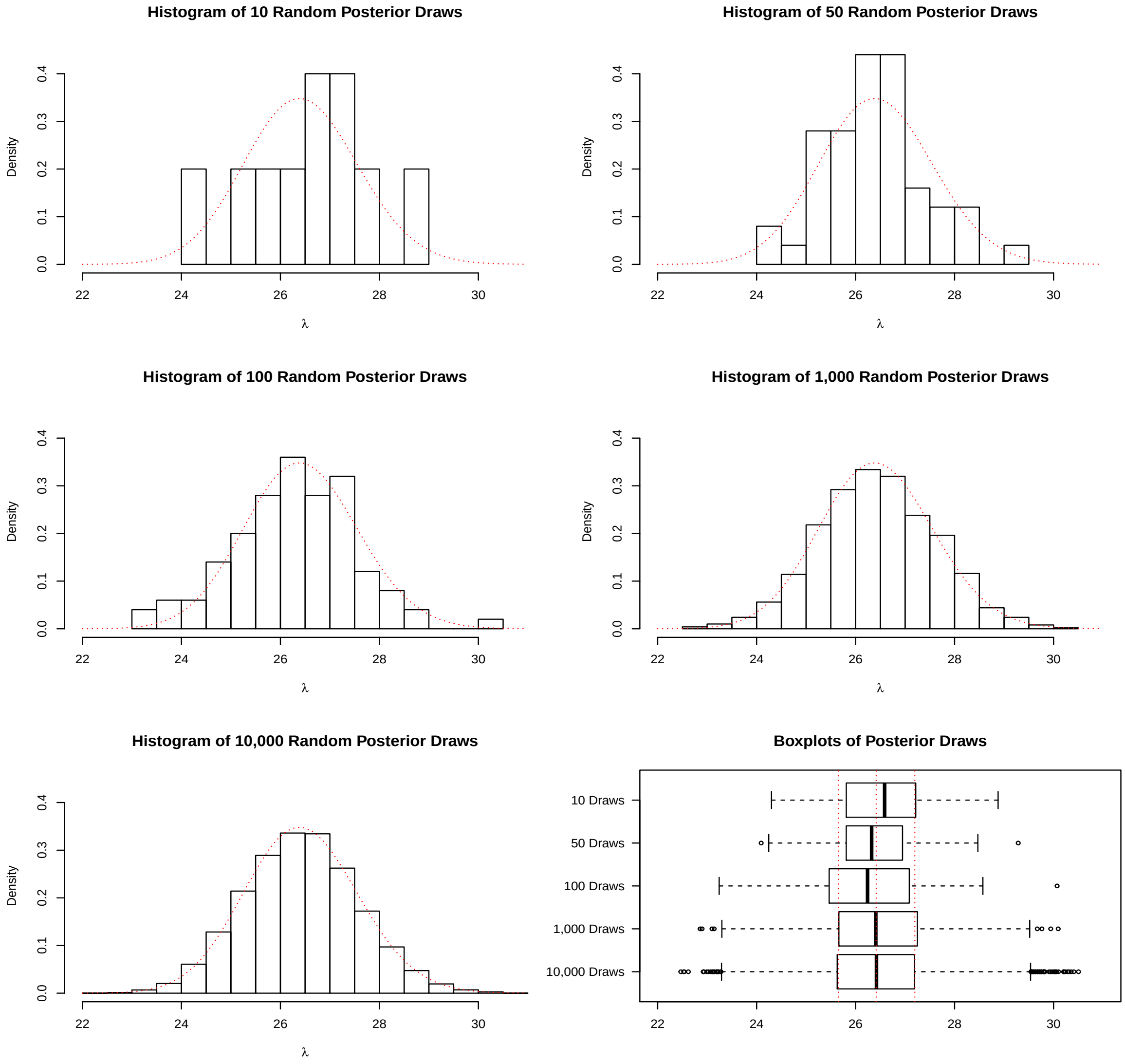
Stat 532 Assignment 4

Kenny Flagg

September 25, 2015

Problem 1

1. The posterior distribution was Gamma(531.3, 20.1). Histograms of posterior draws appear below with the density function overlaid. The vertical lines on the boxplot show the theoretical quartiles.



Histogram of 10 Random Posterior Draws

Histogram of 50 Random Posterior Draws

Histogram of 100 Random Posterior Draws

Histogram of 1,000 Random Posterior Draws

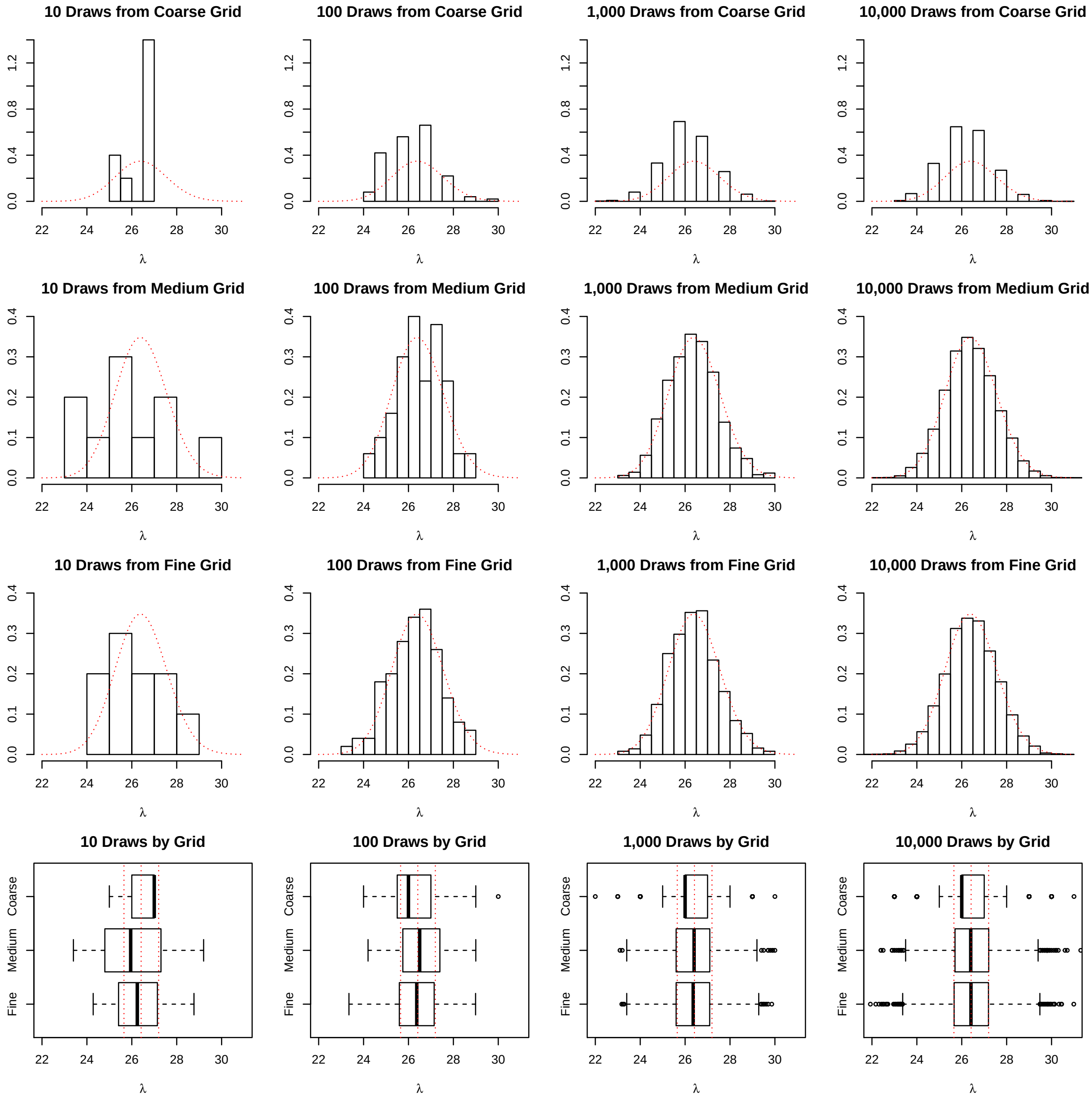
Histogram of 10,000 Random Posterior Draws

Boxplots of Posterior Draws

As I would expect, the distribution of draws looks more like the density curve as the number of draws increases. The simulated results from the sample of size 10,000 agree very closely with the analytic results, as seen in the above plot and in the table below, which shows the posterior interval and probabilities found in the previous assignment.

	90% Posterior Interval	$Pr(10 < \lambda < 20 \mid y)$	$Pr(\lambda < 5)$
Analytic	[24.58, 28.35]	4.4528E-10	1.5880E-199
10,000 Simulated Draws	[24.56, 28.35]	0.0000E+00	0.0000E+00

2. I created grids in the interval $(0, 100]$ and I designated the grids as coarse (100 points spaced 1 unit apart), medium (1,000 points with 0.1 unit spacing), and fine (10,000 points, 0.01 unit spacing). I weighted the points by the associated values of the Gamma(531.3, 20.1) density function and then generated samples of sizes 10, 100, 1,000, and 10,000.



10 Draws from Coarse Grid

100 Draws from Coarse Grid

1,000 Draws from Coarse Grid

10,000 Draws from Coarse Grid

10 Draws from Medium Grid

100 Draws from Medium Grid

1,000 Draws from Medium Grid

10,000 Draws from Medium Grid

10 Draws from Fine Grid

100 Draws from Fine Grid

1,000 Draws from Fine Grid

10,000 Draws from Fine Grid

10 Draws by Grid

100 Draws by Grid

1,000 Draws by Grid

10,000 Draws by Grid

When the goal is to get a good approximation of the distribution, the fineness of the grid is more important than the number of draws.

The coarse grid only had 8 values that appeared in the samples. The size 1,000 and 10,000 samples from the coarse grid had the correct shape to their distributions, but the boxplot shows a discrepancy between the sample quartiles and the theoretical quartiles. There were not enough distinct values to compute precise quantiles and probabilities.

When compared across samples from the same grid, the samples of size 100 and above all had about the same shape. However, the boxplots show that samples of size 100 had different shifts compared to the theoretical quantiles; the distribution seemed unstable, but the instability was reduced by using a finer grid.

I conclude that, when simulating a continuous distribution by a grid approximation, the grids should be as fine as possible, and it is desirable to make at least 1,000 draws.

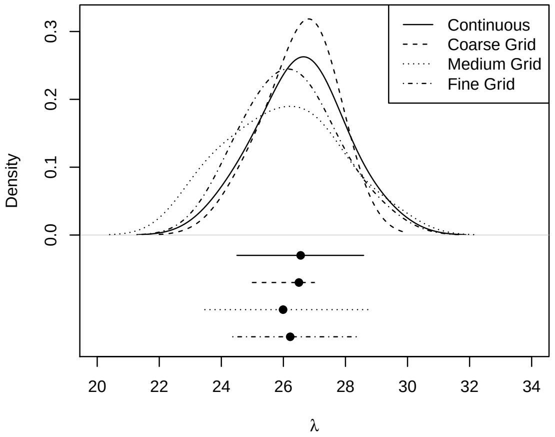
The following function constructed the grids and performed the weighting and sampling. I looped through the grid sizes and sample sizes, calling the function at each iteration.

```r
# Function to find weights from the Gamma(531.3, 20.1) pdf and do
# sampling for specified a number of grid points and sample size.
grid.sample <- function(n.grids, n.samples){
  grid <- seq(0, 100, 100 / n.grids)[-1] # Don't include 0
  weights <- dgamma(grid, 531.3, 20.1)
  samples <- sample(grid, n.samples, replace = TRUE, prob = weights / sum(weights))
  return(samples)
}
```

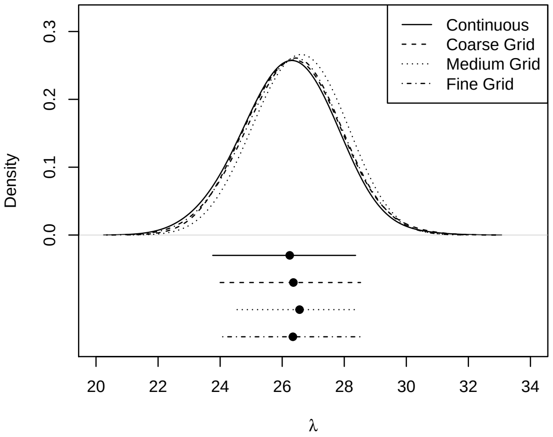
3. Kernel density estimates are a clean way to summarize several empirical distributions on one plot. The plots on the next page show density estimates of the posterior draws using Gaussian kernels with bandwidth 1. The lower portion of each plot show a 95% posterior interval and a point for the posterior mean from each sample of draws. I use “continuous” to refer to the `rgamma()` draws because I think of them as draws from a continuous grid.

   When the sample size was at least 100, the distributions all had similar centers and spreads. For samples sizes of 1,000 and 10,000, the density curves and posterior intervals are barely distinguishable between the medium grid, fine grid, and continuous case. I have become convinced that grid approximations are a reasonable way to simulate continuous distributions.

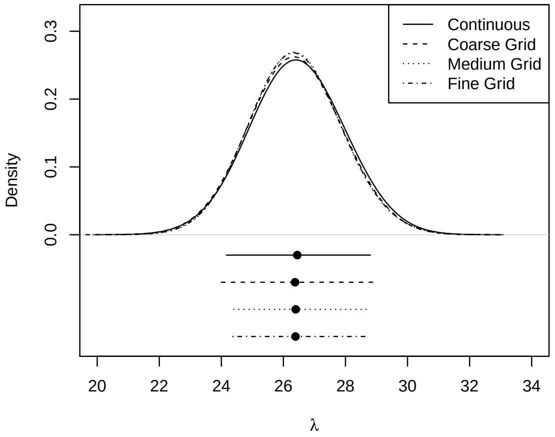
Emperical Density Curves and 95% Intervals from 10 Posterior Draws



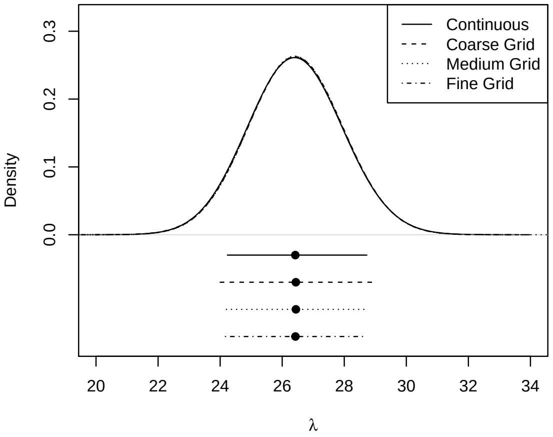
Emperical Density Curves and 95% Intervals from 100 Posterior Draws



Emperical Density Curves and 95% Intervals from 1,000 Posterior Draws



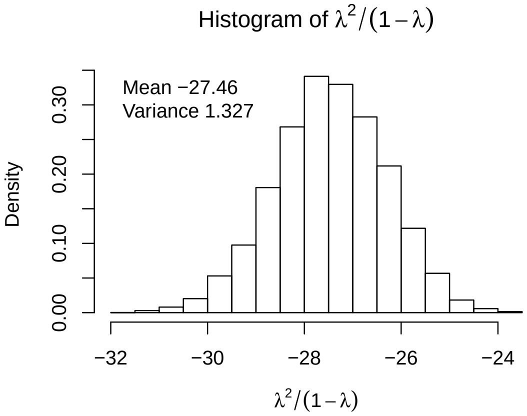
Emperical Density Curves and 95% Intervals from 10,000 Posterior Draws

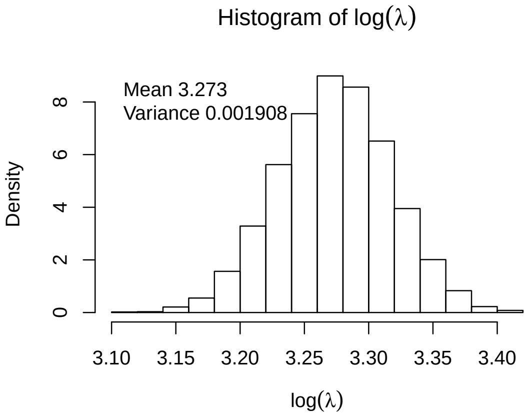


4. I disagree with Gelman's statement that only 100 draws are needed to characterize a distribution. When I generated 100 draws from the theoretical distribution, the distribution of the draws was shifted sightly to the left. The histogram is smooth enough to give a good picture of the shape, but it could be better. Because of the shift, estimates and probabilities from these draws will not be accurate.

   I got more accurate results with larger numbers of draws. My samples of sizes 1,000 and 10,000, had accurate quartiles and the histograms had the correct shape. Since computers make it simple to generate much larger numbers of draws, I think we should use many more than only 100 posterior draws to make inference.

5. I transformed the 10,000 draws from `rgamma()` and the results are shown in the histogram on the next page. Running these simulations sure was easier than using the delta method!





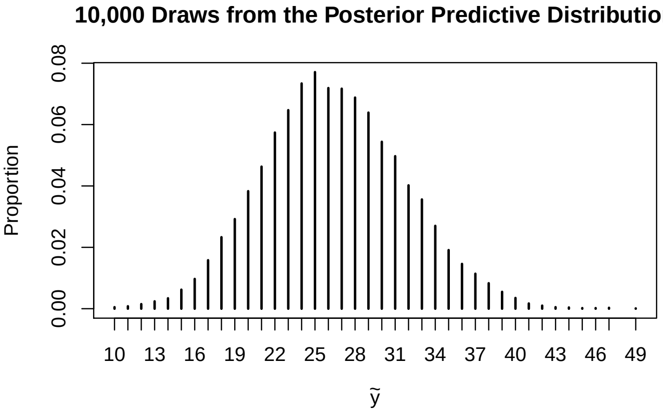
6. The probability mass function of the posterior predictive distribution is

$$\begin{aligned}
p(\tilde{y}|y) &= \int_\lambda p(\tilde{y}|\lambda)p(\lambda|y)d\lambda \\
&= \int_\lambda \frac{\lambda^{\tilde{y}}e^{-\lambda}}{\tilde{y}!}\frac{20.1^{531.3}}{\Gamma(531.3)}\lambda^{530.3}e^{-20.1\lambda}d\lambda \\
&= \frac{20.1^{531.3}}{\Gamma(531.3)\tilde{y}!}\int_\lambda \lambda^{530.3+\tilde{y}}e^{-21.1\lambda}d\lambda \\
&= \frac{\Gamma(531.3+\tilde{y})}{\Gamma(531.3)\tilde{y}!}\frac{20.1^{531.3}}{21.1^{531.3+\tilde{y}}} \\
&= \frac{\Gamma(531.3+\tilde{y})}{\Gamma(531.3)\tilde{y}!}\left(\frac{20.1}{21.1}\right)^{531.3}\left(\frac{1}{21.1}\right)^{\tilde{y}}, \tilde{y} = 0, 1, 2, \ldots
\end{aligned}$$

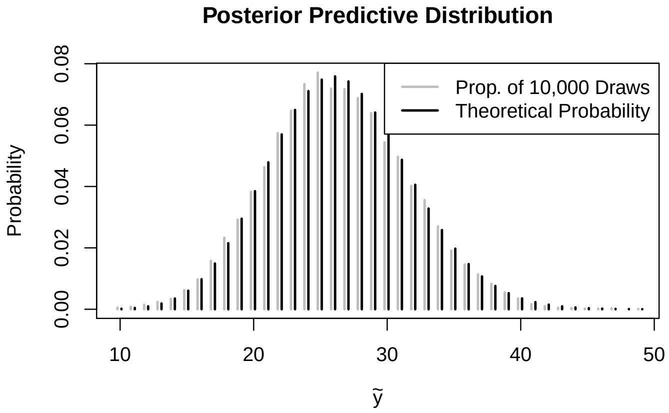
so $\tilde{y}|y \sim \text{NegBin}(531.3, 20.1)$ in Gelman's parameterization.

7. We need to assume that $\tilde{y}$ is exchangeable with the previous observations y_i. That is, $\tilde{y}$ came from the same process or population as the y_i and could have been included in the original data without requiring a different form for the likelihood.

8. I generated 10,000 draws from the NegBin(531.3, 20.1) distribution and plotted them in what is certainly not a histogram.



9. This next plot shows the bars for the random draws along with bars for the values of the probability mass function.



10. The posterior distribution is $\lambda|y \sim \text{Gamma}(531.3, 20.1)$ with

$$E(\lambda|y) = \frac{531.3}{20.1} = 26.433$$

and

$$Var(\lambda|y) = \frac{531.3}{20.1^2} = 1.315.$$

The posterior predictive distribution is $\tilde{y}|y \sim \text{NegBin}(531.3, 20.1)$ with

$$\begin{aligned} E(\tilde{y}|y) &= E(E(\tilde{y}|\lambda)|y) \\ &= E(\lambda|y) \\ &= 26.433 \end{aligned}$$

and

$$\begin{aligned} Var(\tilde{y}|y) &= E(Var(\tilde{y}|\lambda)|y) + Var(E(\tilde{y}|\lambda)|y) \\ &= E(\lambda|y) + Var(\lambda|y) \\ &= 26.433 + 1.315 \\ &= 27.748. \end{aligned}$$

The means of the posterior distribution and the posterior predictive distribution are equal, but the posterior predictive distribution has a larger variance than the posterior distribution. This is because the posterior predictive distribution must account for both the uncertainty in estimating the mean, and the additional uncertainty that comes from individual observations varying about the mean. This is another case of what we've seen in previous courses.

Introductory statistics courses cover the simplest inferential situation, where

$$y_i = \mu + \epsilon_i, \epsilon \sim \text{N}(0, \sigma^2)$$

and if σ^2 is unknown it is estimated by the sample variance s^2. Two standard error formulas are taught for this situation. For a confidence interval or hypothesis test about μ, the formula is

$$SE(\hat{\mu}) = \sqrt{s^2/n}.$$

For a confidence interval about a single observation, we are taught to use

$$SE(\hat{y}_i) = \sqrt{s^2 + s^2/n}.$$

This comes from the fact that we estimated $\hat{\mu}$ from y and so the prediction variance is

$$Var(\tilde{y}|y) = Var(\hat{\mu} + \epsilon) = Var(\hat{\mu}) + Var(\epsilon) = \sigma^2/n + \sigma^2$$

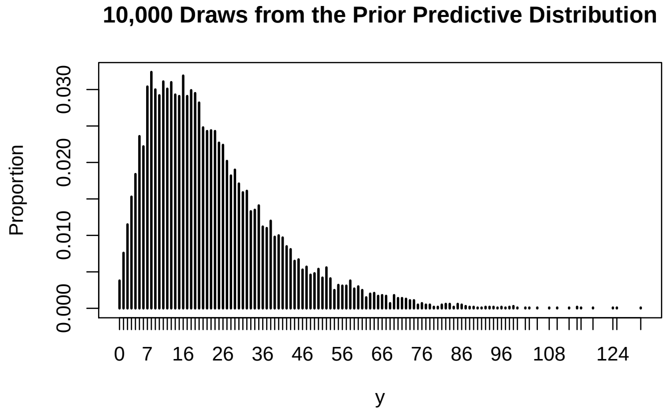
because $\hat{\mu}$ was estimated with uncertainty and ϵ acts as a random adjustment that causes $\tilde{y}$ to vary around the mean.

In Gamma-Poisson problem, we cannot break up the observed value into the sum of a mean and an adjustment. However, we assume that if λ was known then $\tilde{y}| \sim$ Poisson(λ). Since λ is unknown, we model it with a Gamma distribution, and then to make inference about $\tilde{y}$ we must condition on λ. By applying the conditional variance formula, we see that the predictive variance does take the form of a sum of variation from two sources. The variation in estimating λ is incorporated by the $Var(E(\tilde{y}|\lambda)|y)$ term. To this, we add $E(Var(\tilde{y}|\lambda)|y)$, the variation that comes from the Poisson model for $\tilde{y}$.

11. The prior distribution was $\lambda \sim$ Gamma$(2.3, 0.1)$ with $E(\lambda) = 23$ and $Var(\lambda) = 230$. Then the probability mass function of the prior predictive distribution is

$$\begin{aligned}
p(y) &= \int_\lambda p(y|\lambda)p(\lambda)d\lambda \\
&= \int_\lambda \frac{\lambda^y e^{-\lambda}}{y!} \frac{0.1^{2.3}}{\Gamma(2.3)} \lambda^{1.3} e^{-0.1\lambda} d\lambda \\
&= \frac{0.1^{2.3}}{\Gamma(2.3)y!} \int_\lambda \lambda^{1.3+y} e^{-1.1\lambda} d\lambda \\
&= \frac{\Gamma(2.3+y)}{\Gamma(2.3)y!} \frac{0.1^{2.3}}{1.1^{2.3+y}} \\
&= \frac{\Gamma(2.3+y)}{\Gamma(2.3)y!} \left(\frac{0.1}{1.1}\right)^{2.3} \left(\frac{1}{1.1}\right)^y, y = 0, 1, 2, \ldots
\end{aligned}$$

and thus $y \sim$ NegBin$(2.3, 0.1)$. I generated 10,000 draws from this distribution and summarized them with the following plot.



Both the prior predictive distribution and the posterior predictive distribution follow the pattern that if $y|\lambda \sim \text{Poisson}(\lambda)$ and $\lambda \sim \text{Gamma}(\alpha, \beta)$ then $y \sim \text{NegBin}(\alpha, \beta)$.

The prior predictive mean and variance are

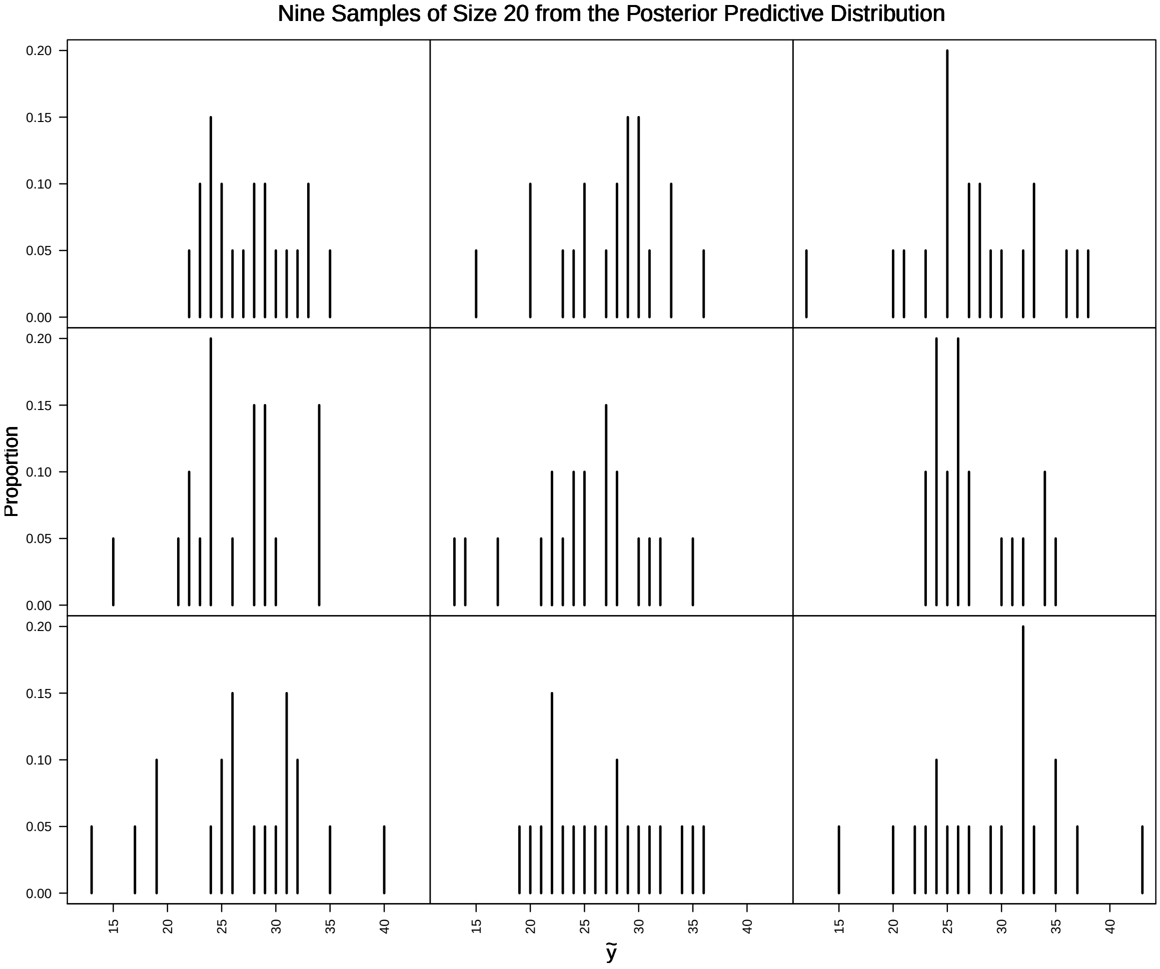
$$\begin{aligned} E(y) &= E(E(y|\lambda)) \\ &= E(\lambda) \\ &= 23 \end{aligned}$$

and

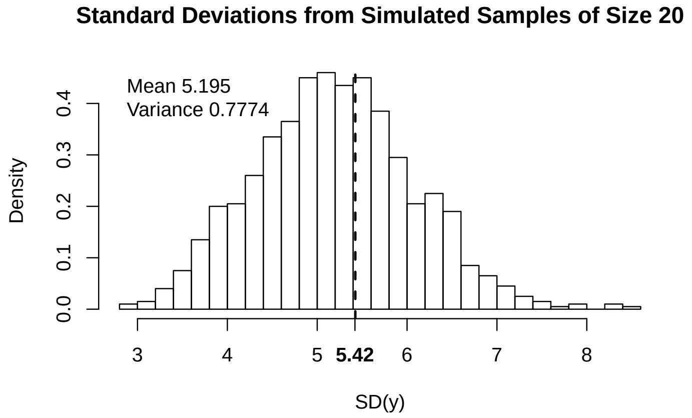
$$\begin{aligned} Var(y) &= E(Var(y|\lambda)) + Var(E(y|\lambda)) \\ &= E(\lambda) + Var(\lambda) \\ &= 23 + 230 \\ &= 253. \end{aligned}$$

The prior predictive distribution has a large variance because it was based on a weak prior. Including the observed data caused the posterior predictive mean to move close to the mean of the data, which was 26.45, and reduced the prediction variance by nearly a factor of 10.

12. Here I have plotted nine samples from the posterior predictive distribution.

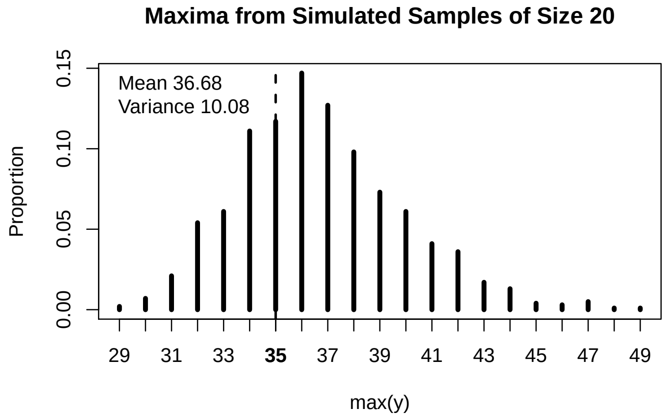


13. I previously "observed" a sample of size 20 with a standard deviation of 5.424. I computed the standard deviations of 1,000 simulated samples from the posterior predictive distribution and plotted the distribution.



The standard deviation of the original sample is right in the middle of this distribution, just where I expected it to be. It is a little to the right of the distribution mean, but close enough that it does not look at all unusual.

14. The original sample had a maximum of 35. I found the maximum values of each of the 1,000 simulated samples and plotted their distribution.



As with the observed standard deviation, the observed maximum is in the center of the distribution near the mean.

Problem 2

Gelman's point here is that a weakly informative prior that contains the most basic information available does not need to look flat.

Consider a linear regression model for the log-transformed heights of trees of a certain species in various locations around the United States. Suppose one predictor is the concentration of a certain mineral in the soil at each location, and this predictor has an approximate normal distribution and we standardize it using the observed mean and standard deviation. Since this variable is normally distributed, it is certain that the data set contains values more than one standard deviation away from the average mineral concentration.

The coefficient for standardized mineral concentration gives the difference in log-transformed height for two trees growing in soil where the concentration is one standard deviation apart. Suppose a tree growing in soil with the average concentration has a (back-transformed) predicted height of 20m, and the mineral concentration coefficient is 10. Then another tree with the same values for all predictors, except that the concentration is one standard deviation higher than the average, will have a predicted height of $20\text{m} \times e^{10} = 440{,}529\text{m}$. This is a preposterous value for the height of a tree. In this case, it would be natural to give the mineral concentration coefficient a prior that constrains it to be much less than 10. Doing so involves only very general knowledge about reasonable heights for trees and will not have an excessive effect on the results.

Problem 3

1. We assume $p(y|\theta) = \theta e^{-\theta y}$, $y > 0$, and $p(\theta) = \frac{\beta^\alpha}{\Gamma(\alpha)}\theta^{\alpha-1}e^{-\beta\theta}$, $\theta > 0$. We need to find

$$p(\theta|y \geq 100) = \frac{p(y \geq 100|\theta)p(\theta)}{p(y \geq 100)}.$$

First,

$$p(y \geq 100|\theta) = \int_{100}^{\infty} \theta e^{-\theta y} dy = e^{-100\theta}.$$

Then,

$$\begin{aligned} p(y \geq 100) &= \int_{100}^{\infty}\int_{0}^{\infty} \theta e^{-\theta y}\frac{\beta^\alpha}{\Gamma(\alpha)}\theta^{\alpha-1}e^{-\beta\theta}d\theta dy \\ &= \frac{\beta^\alpha}{\Gamma(\alpha)}\int_{100}^{\infty}\int_{0}^{\infty} \theta^\alpha e^{-(\beta+y)\theta}d\theta dy \\ &= \frac{\beta^\alpha}{\Gamma(\alpha)}\int_{100}^{\infty}\frac{\Gamma(\alpha+1)}{(\beta+y)^{\alpha+1}}dy \\ &= \alpha\beta^\alpha\int_{100}^{\infty}(\beta+y)^{-\alpha-1}dy \\ &= -\beta^\alpha(\beta+y)^{-\alpha}\Big|_{y=100}^{\infty} \\ &= \left(\frac{\beta}{\beta+100}\right)^\alpha. \end{aligned}$$

Finally,

$$\begin{aligned} p(\theta|y \geq 100) &= \frac{e^{-100\theta}\frac{\beta^\alpha}{\Gamma(\alpha)}\theta^{\alpha-1}e^{-\beta\theta}}{\left(\frac{\beta}{\beta+100}\right)^\alpha} \\ &= \frac{(\beta+100)^\alpha}{\Gamma(\alpha)}\theta^{\alpha-1}e^{-(\beta+100)\theta} \end{aligned}$$

and so $\theta|y \geq 100 \sim \text{Gamma}(\alpha, \beta+100)$. The mean is

$$E(\theta|y \geq 100) = \frac{\alpha}{\beta+100} >$$

and the variance is

$$Var(\theta|y \geq 100) = \frac{\alpha}{(\beta+100)^2}.$$

2. When the data is a single point $y = 100$, I feel comfortable using proportionality shortcuts. So,

$$\begin{aligned} p(\theta|y=100) &= \frac{p(y=100|\theta)p(\theta)}{p(y=100)} \\ &\propto \theta e^{-100\theta}\theta^{\alpha-1}e^{-\beta\theta} \\ &= \theta^{\alpha}e^{-(\beta+100)\theta} \end{aligned}$$

and thus $\theta|y = 100 \sim \text{Gamma}(\alpha+1, \beta+100)$. This has mean

$$E(\theta|y=100) = \frac{\alpha+1}{\beta+100} > E(\theta|y \geq 100)$$

and variance

$$Var(\theta|y=100) = \frac{\alpha+1}{(\beta+100)^2} > Var(\theta|y \geq 100).$$

3. The results are surprising because the posterior distribution of θ has smaller mean and variance when $y \geq 100$ is observed than when $y = 100$ is observed. I expected the mean to be larger when $y \geq 100$ to account for the unboundedness of the observation, and I expected the variance to be larger when $y \geq 100$ because that observation seems to be less informative than when $y = 100$.

Problem 4

The main idea of the *Willful Ignorance* excerpt is that a probability statement simultaneously reflects both knowledge and a lack of knowledge, and so is never absolutely true or false. People frequently misunderstand this, which I have observed in both the professional world and in teaching.

Consider first the client who wishes to make a statement with 99% confidence that his site is some percent free of unexploded ordnance. He has collected enough data to state that the site is 99% clear, but is unsatisfied because it is impractical to perform enough additional sampling to state that it is 99.9% clear. He does not seem to understand that the confidence level is a caveat that uncertainty is unavoidable. The proper response to this confidence statement is not to collect more data to make a safer-sounding statement at the same confidence level, but to treat it as evidence and weigh its consequences when making further decisions.

Teaching has also provided interesting snapshots of how people think about randomness. One of the Stat 216 TEAL activities has students look at sequences of heads and tails, and try to identify which sequences were generated by coins and which were generated by humans. This always leads to a good discussion, and a common observation is that human-generated sequences tend to lack long runs of one outcome, while the coin-generated sequences occasionally have these runs. The students forget that improbable events do happen.

In my experience, people get too interested in attractive numbers, but they do not realize that there is an important difference between improbable (small but nonzero probability) and impossible (probability of 0). Probability is best used to guide informed decision-making in the presence of incomplete knowledge rather than being trusted outright.

R Code Appendix

Problem 1

1.
```
set.seed(98374)
draws10 <- rgamma(10, 531.3, 20.1)
draws50 <- rgamma(50, 531.3, 20.1)
draws100 <- rgamma(100, 531.3, 20.1)
draws1000 <- rgamma(1000, 531.3, 20.1)
draws10000 <- rgamma(10000, 531.3, 20.1)

# Better to put these in a list.
# Reverse order to make the boxplots have smallest on top, largest on bottom.
draws <- list('10,000 Draws' = draws10000,
              '1,000 Draws' = draws1000,
              '100 Draws' = draws100,
              '50 Draws' = draws50,
              '10 Draws' = draws10)
sizes <- c(10000, 1000, 100, 50, 10)

par(mfrow = c(3, 2))
for(i in 5:1){
  hist(draws[[i]], breaks = 15, freq = FALSE,
       xlim = c(22, 31), ylim = c(0, 0.45), xlab = expression(lambda),
       main = paste('Histogram of', prettyNum(sizes[i], big.mark = ','),
                    'Random Posterior Draws'))
  curve(dgamma(x, 531.3, 20.1), col = 'red', lty = 3, add = TRUE)
}
boxplot(draws, horizontal = TRUE, las = 1, ylim = c(22, 31),
        main = 'Boxplots of Posterior Draws')
abline(v = qgamma(c(0.25, 0.5, 0.75), 531.3, 20.1), col = 'red', lty = 3)
```

```
# Maybe this will save me some effort?
interval <- function(x, sig = 4){
  return(paste('[', signif(x[1], sig), ', ', signif(x[2], sig), ']', sep = ''))
}

interval.analytic <- interval(qgamma(c(0.05, 0.95), 531.3, 20.1))
interval.sim <- interval(quantile(draws10000, c(0.05, 0.95)))

prob1.analytic <- pgamma(20, 531.3, 20.1) - pgamma(10, 531.3, 20.1)
prob1.sim <- mean(draws10000 >= 10 & draws10000 < 20)

prob2.analytic <- pgamma(5, 531.3, 20.1)
prob2.sim <- mean(draws10000 < 5)

results <- data.frame(post.interval = c(interval.analytic, interval.sim),
                 prob1 = c(prob1.analytic, prob1.sim),
                 prob2 = c(prob2.analytic, prob2.sim))
colnames(results) <- c('90\\% Posterior Interval',
                       '\\(Pr(10<\\lambda<20|y)\\)',
                       '\\(Pr(\\lambda<5)\\)')
rownames(results) <- c('Analytic', '10,000 Simulated Draws')
print(xtable(results, digits = c(0, 0, -4, -4)),
      sanitize.colnames.function = function(x){x}, floating = FALSE)
```

2.

```r
# Function to find weights from the Gamma(531.3, 20.1) pdf and do
# sampling for specified a number of grid points and sample size.
grid.sample <- function(n.grids, n.samples){
  grid <- seq(0, 100, 100 / n.grids)[-1] # Don't include 0
  weights <- dgamma(grid, 531.3, 20.1)
  samples <- sample(grid, n.samples, replace = TRUE, prob = weights / sum(weights))
  return(samples)
}
```

```r
# Loop through each combination of grid size and sample size.
grid.sizes <- c(100, 1000, 10000)
grid.levels <- c('Coarse Grid', 'Medium Grid', 'Fine Grid')
sample.sizes <- c(10, 100, 1000, 10000)

# The samples will be stored in a matrix of lists to allow differing lengths.
lambda.samples <- matrix(list(),
                    nrow = length(grid.sizes), ncol = length(sample.sizes))

set.seed(123756)
for(i in 1:length(grid.sizes)){
  for(j in 1:length(sample.sizes)){
    lambda.samples[[i, j]] <- grid.sample(grid.sizes[i], sample.sizes[j])
  }
}

# Use 5 bins for sample size 10 and 15 bins for larger samples.
bins <- c(5, 15, 15, 15)

# Use different, appropriate vertical scales for the different grid sizes.
ymax <- c(1.4, 0.4, 0.4)

par(mfrow = c(4, 4), mar = c(4, 3, 3, 1))
for(i in 1:length(grid.sizes)){
  for(j in 1:length(sample.sizes)){
    hist(lambda.samples[[i, j]], xlim = c(22, 31), ylim = c(0, ymax[i]),
         freq = FALSE, breaks = bins[j], xlab = expression(lambda),
         main = paste(prettyNum(sample.sizes[j], big.mark = ','),
                      'Draws from', grid.levels[i]))
    curve(dgamma(x, 531.3, 20.1), col = 'red', lty = 3, add = TRUE)
  }
}
for(j in 1:length(sample.sizes)){
  boxplot(list('Fine' = lambda.samples[[3, j]],
               'Medium' = lambda.samples[[2, j]],
               'Coarse' = lambda.samples[[1, j]]),
          horizontal = TRUE, ylim = c(22, 31), xlab = expression(lambda),
          main = paste(prettyNum(sample.sizes[j], big.mark = ','),
                       'Draws by Grid'))
  abline(v = qgamma(c(0.25, 0.5, 0.75), 531.3, 20.1), col = 'red', lty = 3)
}
```

3.

```r
# Put the desired draws in a list.
# Change order back to 10, 100, 1000, 10000.
lambda.draws <- draws[c(5, 3, 2, 1)]

par(mfrow = c(2, 2))
```

```
# Loop for sample sizes.
for(j in 1:length(sample.sizes)){
  # Density curve from continuous support.
  plot(density(lambda.draws[[j]], bw = 1),
       xlim = c(20, 34), ylim = c(-0.16, 0.32), yaxt = 'n', lty = 1,
       xlab = expression(lambda),
       main = paste('Emperical Density Curves and 95% Intervals\nfrom',
                    prettyNum(sample.sizes[j], big.mark = ','),
                    'Posterior Draws'))

  # 95% posterior interval and mean from continuous support.
  lines(x = quantile(lambda.draws[[j]], c(0.025, 0.975)),
        y = rep(-0.03, 2), lty = 1)
  points(x = mean(lambda.draws[[j]]), y = -0.03, type = 'p', pch = 19)

  # Loop for grid sizes.
  for(i in 1:length(grid.sizes)){
    # Density curves from grids.
    lines(density(lambda.samples[[i, j]], bw = 1), lty = i+1)

    # 95% posterior interval and mean.
    lines(x = quantile(lambda.samples[[i, j]], c(0.025, 0.975)),
          y = rep(-0.04, 2)*(i+1)+0.01, lty = i+1)
    points(x = mean(lambda.samples[[i, j]]), y = -0.04*(i+1)+0.01,
           type = 'p', pch = 19)
  }

  # Add a legend.
  legend('topright', lty = 1:4,
         legend = c('Continuous', grid.levels))
  axis(2, at = seq(0, 0.3, 0.1))
}
```

5.
```
trans1 <- draws10000^2 / (1-draws10000)
trans2 <- log(draws10000)

par(mfrow = c(1, 2))
hist(trans1, freq = FALSE, xlab = expression(lambda^2/(1-lambda)),
     main = expression(paste('Histogram of ', lambda^2/(1-lambda))))
text(x = -32, y = 0.3, pos = 4,
     labels = paste('Mean', signif(mean(trans1), 4),
                    '\nVariance', signif(var(trans1), 4)))

hist(trans2, freq = FALSE, xlab = expression(log(lambda)),
     main = expression(paste('Histogram of ', log(lambda))))
text(x = 3.1, y = 7.8, pos = 4,
     labels = paste('Mean', signif(mean(trans2), 4),
                    '\nVariance', signif(var(trans2), 4)))
```

8.
```
set.seed(821924)
ytilde <- rnbinom(10000, 531.3, 20.1/21.1)

plot(table(ytilde)/10000, xlab = expression(tilde(y)), ylab = 'Proportion',
     main = '10,000 Draws from the Posterior Predictive Distribution')
```

9.
```r
ys <- min(ytilde):max(ytilde)

plot(x = as.numeric(rownames(table(ytilde)))-0.2,
     y = table(ytilde)/10000, type = 'h', lwd = 2, col = 'grey',
     xlab = expression(tilde(y)), ylab = 'Probability',
     main = 'Posterior Predictive Distribution')
points(x = ys+0.1, y = dnbinom(ys, 531.3, 20.1/21.1),
       type = 'h', lwd = 2, col = 'black')
legend('topright', lwd = 2, col = c('grey', 'black'),
       legend = c('Prop. of 10,000 Draws', 'Theoretical Probability'))

# Why doesn't plot() give me a vertical axis?
axis(2, at = pretty(c(0, 0.08)))
```

11.
```r
set.seed(456734)
priory <- rnbinom(10000, 2.3, 0.1/1.1)

plot(table(priory)/10000, xlab = expression(y), ylab = 'Proportion',
     main = '10,000 Draws from the Prior Predictive Distribution')
```

12.
```r
set.seed(7823)
twentysamples <- matrix(rnbinom(20000, 531.3, 20.1/21.1), ncol = 20)

# Randomly choose some samples to plot
idx <- sample(1000, 9)

# Get some bounds to set up axes
ymin <- min(twentysamples[idx,])
ymax <- max(twentysamples[idx,])

# I need an upper bound of the proportions, so I'll tabulate each sample,
# get the max count in each table, unlist that, and get that vector's max
# to get the maximum count in any single sample. Then divide that by the
# sample size.
pmax <- max(unlist(lapply(apply(twentysamples[idx,], 1, table), max))) / 20

# ggplot is a cop-out!
par(mfrow = c(3, 3), mar = c(0,0,0,0), oma = c(5, 4, 4, 2))
for(i in 1:length(idx)){
  plot(table(twentysamples[idx[i],]) / 20, axes = FALSE,
       xlim = c(ymin, ymax), ylim = c(0, pmax))

  # Create axes and things when necessary
  if(i %% 3 == 1){
    axis(2, at = pretty(c(0, pmax)), las = 2)
  }
  if(i > 6){
    axis(1, at = pretty(c(ymin, ymax)), las = 2)
  }
  mtext('Nine Samples of Size 20 from the Posterior Predictive Distribution',
        side = 3, line = 1, cex = 1.2, outer = TRUE)
  mtext('Proportion', side = 2, line = 3, outer = TRUE)
  mtext(expression(tilde(y)), side = 1, line = 3, outer = TRUE)
}
```

13.

```
# Reproduce observations from previous assignment.
set.seed(9231346)
lambda <- rgamma(1, 2.3, 0.1)
obs <- rpois(20, lambda)

sds <- apply(twentysamples, 1, sd)
hist(sds, freq = FALSE, breaks = 30, xlab = 'SD(y)',
     main = 'Standard Deviations from Simulated Samples of Size 20')
abline(v = sd(obs), lwd = 2, lty = 2)
axis(1, at = 5.42, font = 2)
text(x = 2.75, y = 0.4, pos = 4,
     labels = paste('Mean', signif(mean(sds), 4),
                    '\nVariance', signif(var(sds), 4)))
```

14.

```
maxs <- apply(twentysamples, 1, max)
plot(table(maxs) / 1000, lwd = 4, xlab = 'max(y)', ylab = 'Proportion',
     main = 'Maxima from Simulated Samples of Size 20')
abline(v = max(obs), lwd = 2, lty = 2)
axis(1, at = 35, font = 2)
text(x = 28.5, y = 0.13, pos = 4,
     labels = paste('Mean', signif(mean(maxs), 4),
                    '\nVariance', signif(var(maxs), 4)))
```