

# GR5065 Homework 2

Ben Goodrich

Due February 21, 2022 at 4PM

## 1 Gross Domestic Product

```
library(readr)
library(dplyr)
FRED <- "https://fred.stlouisfed.org/graph/fredgraph.csv"
dataset <- read_csv(paste0(FRED, "?id=A261RL1Q225SBEA,A191RL1Q225SBEA"), na = ".") %>%
  na.omit %>%
  rename(quarter_startdate = DATE, GDI = A261RL1Q225SBEA, GDP = A191RL1Q225SBEA)
```

### 1.1 Normal Prior

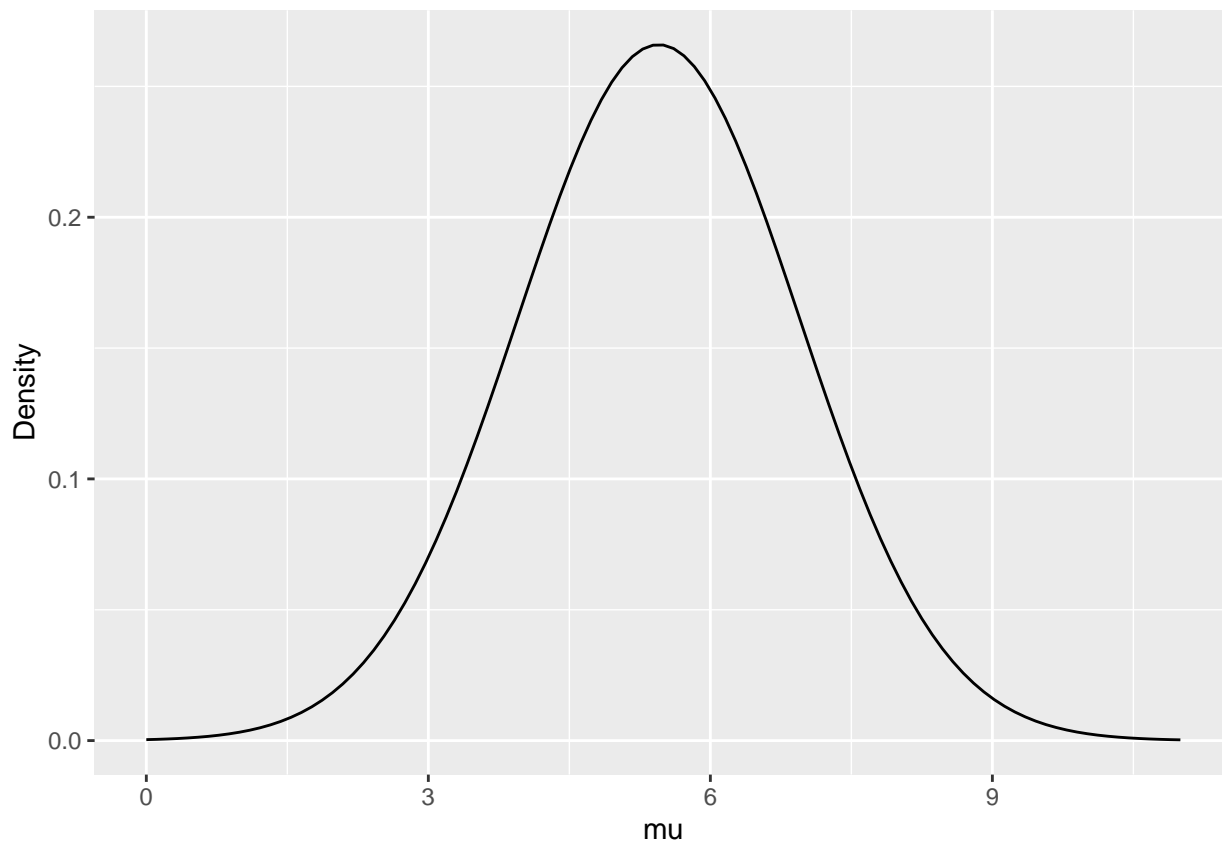
Describe your beliefs about  $\mu$  (the true annualized GDP\* growth) during the fourth quarter of 2021 with a normal distribution that has expectation  $m$  and standard deviation  $s$  (which you should choose values for). And plot it.

You could approach this in various ways. One is to utilize the empirical regularity known as “Okun’s Law” that was mentioned in

[https://obamawhitehouse.archives.gov/sites/default/files/docs/gdo\\_issue\\_brief\\_final.pdf](https://obamawhitehouse.archives.gov/sites/default/files/docs/gdo_issue_brief_final.pdf)

which says that the change in GDP\* above its trend is about double the change in the unemployment rate, which was  $-0.7$ . The economy has been growing above its usual trend since covid vaccines have become available, so let’s construct  $m = 5.45$  as  $-2$  times  $-0.7$ , plus the GDO of the third quarter of 2021, which was  $4.05$ . Then,  $s$  reflects how much GDP\* is likely to deviate from  $m$ . There is about a 2 in 3 chance that GDP\* is between  $m \mp s$ , with a 1 in 6 chance that it is less and a 1 in 6 chance it is greater. So, let’s set  $s = 1.5$ .

```
library(ggplot2)
m <- 5.45
s <- 1.5
ggplot() + xlim(0, 11) + ylab("Density") + xlab("mu") +
  geom_function(fun = dnorm, args = list(mean = m, sd = s))
```



## 1.2 Bivariate Normal Likelihood

Write a likelihood function in R

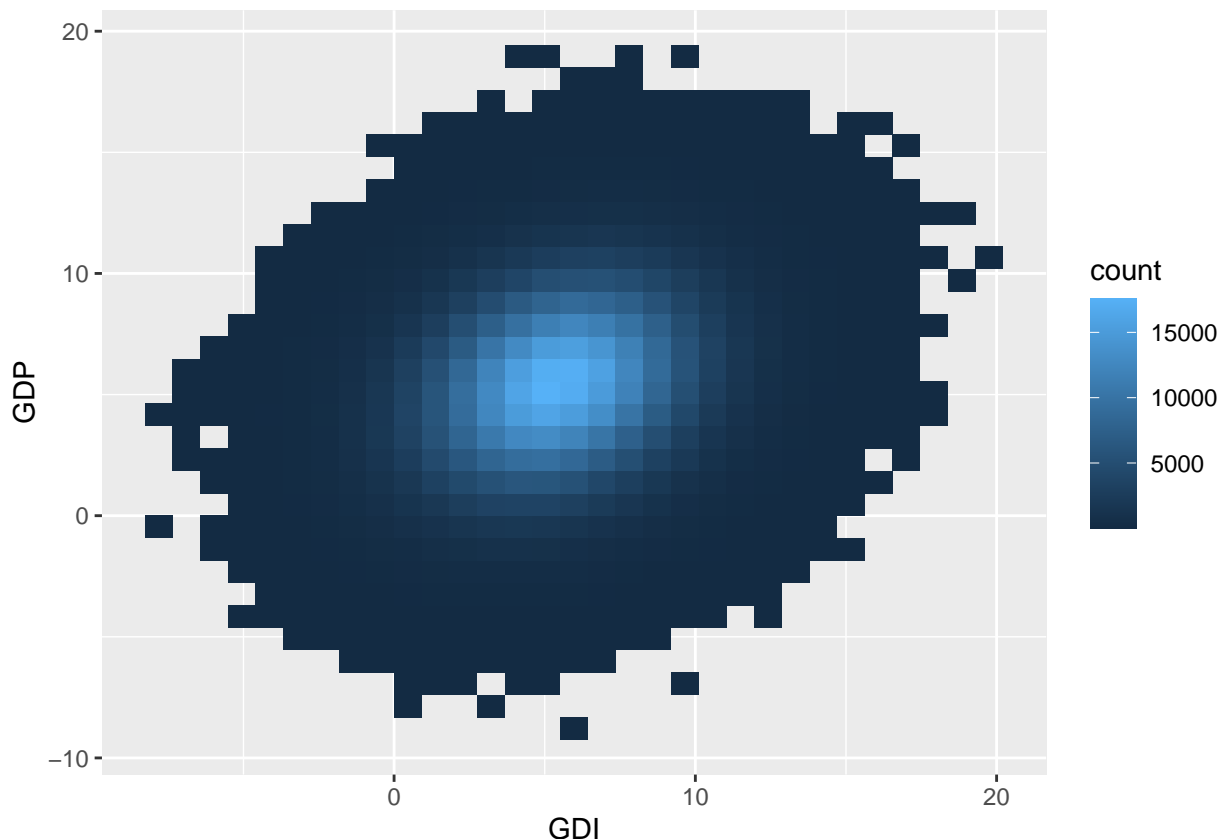
```
sigma <- 7 / 3
rho <- -1 / 10

L <- function(mu, GDI, GDP) {
  # function of mu to be evaluated at observed GDI and GDP only
  dnorm(GDI, mean = mu, sd = sigma) *
  dnorm(GDP, mean = mu + rho * (GDI - mu), sd = sigma * sqrt(1 - rho^2))
}
```

## 1.3 Simulating Reported GDI and GDP Growth

Take one million random draws from your normal prior distribution for  $\mu$  (given  $m$  and  $s$ ). For each of those one million realizations of  $\mu$ , take one random draw of both GDI and GDP growth under the assumptions of the previous subproblem. Create some useful plot of the distribution of what you believe the US government will report for GDI and GDP growth on February 24th. You will notice that the estimated standard deviation over one million draws of each series exceeds  $\frac{7}{3}$ . Explain why.

```
library(tibble)
tibble(mu = rnorm(10^6, mean = m, sd = s),
       GDI = rnorm(10^6, mean = mu, sd = sigma),
       GDP = rnorm(10^6, mean = mu + rho * (GDI - mu),
                   sd = sigma * sqrt(1 - rho^2))) %>%
  ggplot() + geom_bin_2d(aes(x = GDI, y = GDP))
```



This plot exhibits more variation, such that there is some probability that *reported* GDI and / or GDP growth is negative, even though we saw above that the prior probability of the true GDP\* being negative is negligible. This reflects the important fact that our uncertainty about true GDP\* — as captured by  $s$  — is propagated through to our uncertainty about the reported GDI and GDP growth. In other words, we are somewhat uncertain about what GDP\* is, but even if we knew GDP\*, we would be somewhat uncertain about what the government will report for GDI and GDP growth based on its model and incomplete data. These two sources of uncertainty combine when GDI and GDP growth are simulated above via composition.

## 1.4 Posterior

Write a function in R to evaluate the numerator of Bayes rule for the conditional distribution of  $\mu$  given reported GDI and GDP growth (as well as the assumed values for the other parameters). Note that this R function must work with a *vector* of  $\mu$  values and return a vector of the same size containing trivariate density values between  $\mu$ , GDI growth, and GDP growth.

```
numerator <- function(mu) dnorm(mu, mean = m, sd = s) * L(mu, GDI, GDP)
```

Then write code to evaluate the denominator of Bayes rule. Note that you will not be able to actually evaluate this denominator until the values of GDI and GDP growth are revealed on February 24th, but you should write the code now and specify `eval = FALSE` in the header of the R code chunk so that your RMarkdown file will successfully knit to a PDF.

```
GDI <- NA_real_ # we will not know these until 8:30AM on February 24th at https://www.bea.gov/
GDP <- NA_real_
denominator <- integrate(numerator, lower = -Inf, upper = Inf)$value
```

## 1.5 Gross Domestic Output

The paper linked above advocates calculating Gross Domestic Output (GDO), which is just the average of GDI and GDP growth, as a better measure of GDP\* than either GDI or GDP is individually. Figure 4 of the paper also shows that a simple average is close to the optimal weighted average, which would put slightly more weight on GDI growth.

How does GDO conceptually compare and contrast with the posterior distribution of GDP\* conditional on GDI and GDP growth?

GDO is a point estimate that strikes an equal-weighted balance between GDI and GDP growth. The posterior distribution of GDP\* conditional on GDI and GDP growth, as well as  $m$ ,  $s$ ,  $\sigma_I$ ,  $\sigma_P$ , and  $\rho$  strikes a balance between the prior and the to-be-observed data. But more importantly, the posterior distribution also reflects our uncertainty about GDP\* given what we believed before February 24th and the data we (will have) observed after February 24th. Nothing that the government reports on February 24th will have any estimate of uncertainty, even though the government acknowledges that the reported data are highly uncertain and subject to several rounds of revision in the future as more complete data becomes available to put into its model.

Although you did not need to do so for this homework, we can make our answer more explicit by deriving the posterior distribution analytically, which is possible when all of the distributions involved are normal with known standard deviations. [Section 5.3](#) of *Bayes Rules* goes through the math to prove that normal prior distributions are naturally conjugate with univariate normal likelihoods with known standard deviations ( $\sigma_I = \frac{7}{3}$  in this case). The posterior distribution of  $\mu$  conditional on GDI growth ( $I$ ) but not (yet) on GDP growth ( $P$ ) would be a normal distribution with expectation

$$m^* = m \frac{\sigma_I^2}{s^2 + \sigma_I^2} + I \frac{s^2}{s^2 + \sigma_I^2}$$

and standard deviation

$$s^* = s \sigma_I \sqrt{\frac{1}{s^2 + \sigma_I^2}}$$

Thus,  $\mu^*$  would be a weighted average of  $m$  and  $I$ , where the weights are a function of the variances rather than being equiweighted.

In the case of a bivariate normal likelihood, the math is more involved. If we let  $g$  stand for GDI growth (because Mathematica interprets  $I$  as  $\sqrt{-1}$ ),  $G$  for GDP growth, and  $\sigma$  is both  $\sigma_I$  and  $\sigma_P$ , you can execute in Mathematica

```
PDF[NormalDistribution[m, s], \[Mu]] *
PDF[NormalDistribution[\[Mu], \[Sigma]], g] *
PDF[NormalDistribution[\[Mu] + \[Rho] * (g - \[Mu]), \[Sigma] * Sqrt[1 - \[Rho]^2]], G]

Integrate[%, {\[Mu], -Infinity, Infinity},
Assumptions -> {s > 0, \[Sigma] > 0, \[Rho] > -1, \[Rho] < 1}]

FullSimplify[% / %, Reals,
Assumptions -> {s > 0, \[Sigma] > 0, \[Rho] > -1, \[Rho] < 1}]
```

to obtain another univariate normal PDF for the posterior PDF, whose expectation can be recognized as

$$m^{**} = m \frac{(1 + \rho) \sigma^2}{2s^2 + (1 + \rho) \sigma^2} + \text{GDO} \frac{2s^2}{2s^2 + (1 + \rho) \sigma^2}$$

and standard deviation  $s^{**} = s \sigma \sqrt{\frac{1 + \rho}{2s^2 + (1 + \rho) \sigma^2}}$ .

We can verify the analytical and numerical PDFs are the same at any value of  $\mu$  with

```
mu <- 7
c(exact = dnorm(mu, mean = (2 * s^2 * (GDI + GDP) / 2 + m * (1 + rho) * sigma^2) /
                        (2 * s^2 + (1 + rho) * sigma^2),
  sd = s * sigma * sqrt((1 + rho) / (2 * s^2 + (1 + rho) * sigma^2))),
  approximate = numerator(mu) / denominator)
```

As  $s \uparrow \infty$ ,  $m^{**} \rightarrow \text{GDO}$ , so averaging GDI and GDP growth can be seen as a limiting case of Bayesian analysis with a vague prior (and assuming  $\sigma_I = \sigma = \sigma_P$ ). For finite  $s$ , the posterior expectation is again weighted by variances, rather than equiweighting. Moreover, the posterior distribution reflects our uncertainty about GDP\* after having observed the data, whereas GDO does not. Under our assumptions  $s^{**} \approx 1.08$ , which is less than either  $s$  or  $\sigma$  but still rather large substantively.

## 2 Bowling

If your working directory is Assignments/HW2/, execute

```
# defines Pr(x, n = 10, theta = 1) and Omega as 0:10
source(file.path("../", "..", "Week04", "bowling.R"))
```

The `Pr` function is equivalent to

$$\Pr(x \mid n, \theta) = \frac{\log_{n+1+\theta} \left( 1 + \frac{1}{n+\theta-x} \right)}{1 - \log_{n+1+\theta}(\theta)}$$

like we used at the end of Week04. The function body looks different because it tries to be more numerically accurate for small or large values of the inability parameter,  $\theta \geq 0$ , but it should yield the same probabilities as in the mathematical definition above.

### 2.1 Prior

Think about what you believe  $\theta$  to be for a woman who is bowling in the Olympics, if bowling were an Olympic sport (which it is not because it is too boring). In particular, choose an exponential prior for  $\theta$  with expectation  $m$  (different from the  $m$  in the previous problem). Somehow plot your exponential prior in R.

Since

```
sum(Omega * Pr(Omega, n = 10, theta = 0.15))
```

```
## [1] 8.007026
```

is about 8, if we think that elite women bowlers would average around 8 pins knocked down on their first roll of a frame of bowling, then

```
m <- 0.15
```

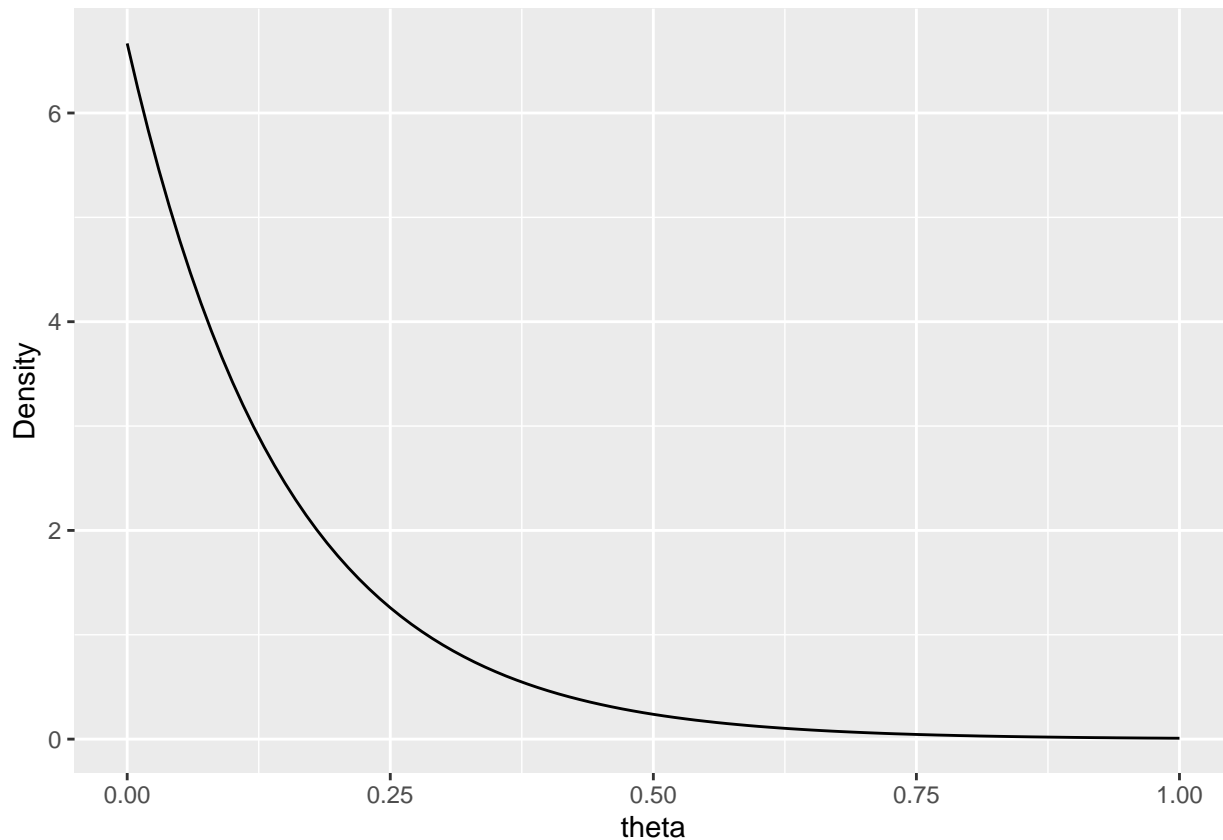
would be reasonable. The conditional expectation of the first roll would actually be

```
E <- function(theta) {
  dexp(theta, rate = 1 / m) *
  sapply(theta, FUN = function(t) sum(Omega * Pr(Omega, n = 10, theta = t)))
}
integrate(E, lower = 0, upper = Inf)
```

```
## 8.142713 with absolute error < 6.4e-06
```

which is fine in this context. This prior also puts considerable probability that  $\theta$  is near 0 but has a long tail (not shown much):

```
ggplot() + xlim(0, 1) + ylab("Density") + xlab("theta") +
  geom_function(fun = dexp, args = list(rate = 1 / m))
```



## 2.2 Marginal Probability of a Frame

Create an  $11 \times 11$  matrix whose cells contain the marginal probability of knocking down  $x_1$  pins on the first roll (the row index) of a frame of bowling and  $x_2$  pins on the second roll (the column index) of that same frame of bowling, irrespective of  $\theta$  but taking into account your exponential prior beliefs about  $\theta$  for this woman. As in class, if the bowler gets a strike on the first roll of a frame, then presume the bowler is guaranteed to knock down zero pins on the second roll of that frame (which does not actually happen in genuine bowling).

What is the sum over all the elements of this matrix, what should the sum be in theory, and what accounts for the difference?

```
marginal_Pr <- matrix(0, nrow = 11, ncol = 11,
  dimnames = list(Omega, Omega))
for (x_1 in Omega) {
  for(x_2 in 0:(10 - x_1)) {
    marginal_Pr[x_1 + 1, x_2 + 1] <- integrate(function(theta) {
      dexp(theta, rate = 1 / m) *
      sapply(theta, FUN = function(t) {
        Pr(x_1, n = 10, theta = t) *
        Pr(x_2, n = 10 - x_1, theta = t)
      })
    }, lower = 0, upper = Inf)$value
  }
}
```

```
round(marginal_Pr, digits = 4)
```

```
##           0           1           2           3           4           5           6           7           8           9          10
## 0  0.0004 0.0005 0.0005 0.0006 0.0007 0.0008 0.0010 0.0013 0.0017 0.0028 0.01
## 1  0.0005 0.0006 0.0007 0.0008 0.0009 0.0011 0.0014 0.0020 0.0032 0.0112 0.00
## 2  0.0007 0.0008 0.0009 0.0011 0.0013 0.0016 0.0023 0.0037 0.0128 0.0000 0.00
## 3  0.0009 0.0010 0.0012 0.0015 0.0019 0.0026 0.0043 0.0149 0.0000 0.0000 0.00
## 4  0.0012 0.0015 0.0018 0.0023 0.0031 0.0051 0.0177 0.0000 0.0000 0.0000 0.00
## 5  0.0018 0.0022 0.0028 0.0038 0.0062 0.0216 0.0000 0.0000 0.0000 0.0000 0.00
## 6  0.0028 0.0035 0.0049 0.0079 0.0276 0.0000 0.0000 0.0000 0.0000 0.0000 0.00
## 7  0.0048 0.0066 0.0107 0.0373 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.00
## 8  0.0100 0.0162 0.0561 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.00
## 9  0.0301 0.1044 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.00
## 10 0.5097 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.00
```

```
sum(marginal_Pr)
```

```
## [1] 1.000004
```

This sum should conceptually be 1 since some combination of  $x_1$  and  $x_2$  must happen on a frame of bowling, irrespective of  $\theta$ . However, it differs in the sixth decimal place because the `integrate` function is not perfectly accurate and entails some numerical error, particularly when integrating functions that have substantial values near the bounds of integration. Such is the price to be paid for being able to obtain the area under the curve of functions that lack elementary antiderivatives, but the discrepancy is minor in this case (and many others).

## 2.3 Data

Once you have completed the previous two subproblems, ask Ben on Ed Discussions for a link to a bowling scorecard for an elite woman bowler. The scorecard will look like

TKM Atayev	1	2	3	4	5	6	7	8	9	10	Total												
Lane 19												Game 1				2019/11/18 9.10				9.43			
Hdcp	0	8	-	7	1		X	9	-	9	/	-	7		X	8	1	7	2	X	9	/	
		8		16		35		44		54		61		80		89		98		118			118
		<div>ooo● oo● oo o</div>		<div>ooo● oo● oo● o</div>		<div>oooo oooo oo o</div>		<div>oooo oooo oo● o</div>		<div>oooo oooo oo o</div>		<div>oooo● oo● oo● oo● o</div>		<div>oooo oooo oo o</div>		<div>oooo oooo● oo● o</div>		<div>oo●o oo● oo● o</div>		<div>oooo oooo oo o</div>		<div>oooo● oooo oo o</div>	

Figure 1: a bowling scorecard

Everyone had a different bowler and hence a different realization of six games of bowling for that bowler. For this answer key, I will use the data generated by Melania Rossi from Italy given at

<https://www.qubicaamf.com/qubicaamf-bowling-world-cup/55bwc/frame-by-frame-files/italy-w-01-06.pdf>

In matrices, these data look like

```
game_1 <- matrix(c(
  7, 2, NA,
  7, 3, NA,
  10, NA, NA,
  9, 1, NA,
  7, 1, NA,
```

```

10, NA, NA,
9, 1, NA,
9, 1, NA,
9, 1, NA,
10, 10, 7),
nrow = 10, ncol = 3, byrow = TRUE)

game_2 <- matrix(c(
6, 3, NA,
10, NA, NA,
7, 3, NA,
5, 3, NA,
7, 2, NA,
6, 3, NA,
8, 1, NA,
7, 2, NA,
9, 1, NA,
10, 9, 1),
nrow = 10, ncol = 3, byrow = TRUE)

game_3 <- matrix(c(
7, 1, NA,
9, 1, NA,
10, NA, NA,
7, 3, NA,
10, NA, NA,
8, 2, NA,
7, 2, NA,
7, 3, NA,
9, 1, NA,
10, 9, 0),
nrow = 10, ncol = 3, byrow = TRUE)

game_4 <- matrix(c(
8, 1, NA,
8, 1, NA,
10, NA, NA,
8, 2, NA,
9, 1, NA,
10, NA, NA,
10, NA, NA,
9, 1, NA,
9, 1, NA,
5, 3, NA),
nrow = 10, ncol = 3, byrow = TRUE)

game_5 <- matrix(c(
10, NA, NA,
8, 2, NA,
9, 1, NA,
10, NA, NA,
9, 1, NA,
6, 3, NA,

```



```

7, 2, NA,
9, 1, NA,
9, 1, NA,
9, 1, 10),
nrow = 10, ncol = 3, byrow = TRUE)

game_6 <- matrix(c(
9, 1, NA,
9, 1, NA,
7, 2, NA,
8, 2, NA,
9, 1, NA,
10, NA, NA,
8, 2, NA,
10, NA, NA,
9, 1, NA,
9, 1, 8),
nrow = 10, ncol = 3, byrow = TRUE)

```

## 2.4 Posteriors

For *your* bowler and prior, obtain the posterior distribution of  $\theta$  given the entire first game of bowling, assuming that frames are independent of each other. This would be similar to what we did at the end of Week03, except now  $\theta$  is a continuous parameter. Plot this posterior distribution of  $\theta$ .

Then, for each of the next five bowling games that you have for your bowler, plot the posterior distribution of  $\theta$  conditioning on all the data *through* that game of bowling. So, there will be six plots in total for this subproblem. Note that it is presumably easier in this case not to use an intermediate posterior distribution as a prior for  $\theta$  when conditioning on the next game. Rather, stick with your original exponential prior for  $\theta$  but condition on multiple games of bowling as you go along. Both approaches will yield the same answer, but the latter is easier to implement. How does the final posterior distribution compare to your prior distribution from the first subproblem?

```

log_likelihood <- function(theta, game) {
  sapply(theta, FUN = function(t) {
    sum(log(Pr(x = game[, 1], n = 10, theta = t)), # first rolls of each frame
        log(Pr(x = game[1:9, 2], n = 10 - game[1:9, 1], theta = t)), # second rolls through frame 9
        # additional roll(s) in frame 10
        log(Pr(x = game[ 10, 2], theta = t,
              # if the bowler got a strike, there are 10 pins upright for the second roll
              n = ifelse(game[10, 1] == 10, 10, 10 - game[10, 1]))),
        log(Pr(x = game[ 10, 3], theta = t,
              # if the bowler got two strikes or a spare, there are 10 pins upright for the third roll
              n = ifelse(game[10, 2] == 10 | (game[10, 1] + game[10, 2]) == 10,
                        10, 10 - game[10, 2]))),
        na.rm = TRUE) # omit rolls that did not actually happen
  })
}

numerator_1 <- function(theta) {
  exp(dexp(theta, rate = 1 / m, log = TRUE) +
      log_likelihood(theta, game = game_1))
}

denominator_1 <- integrate(numerator_1, lower = 0, upper = Inf)$value

```

```

numerator_2 <- function(theta) {
  exp(dexp(theta, rate = 1 / m, log = TRUE) +
    log_likelihood(theta, game = game_1) +
    log_likelihood(theta, game = game_2))
}
denominator_2 <- integrate(numerator_2, lower = 0, upper = Inf)$value

numerator_3 <- function(theta) {
  exp(dexp(theta, rate = 1 / m, log = TRUE) +
    log_likelihood(theta, game = game_1) +
    log_likelihood(theta, game = game_2) +
    log_likelihood(theta, game = game_3))
}
denominator_3 <- integrate(numerator_3, lower = 0, upper = Inf)$value

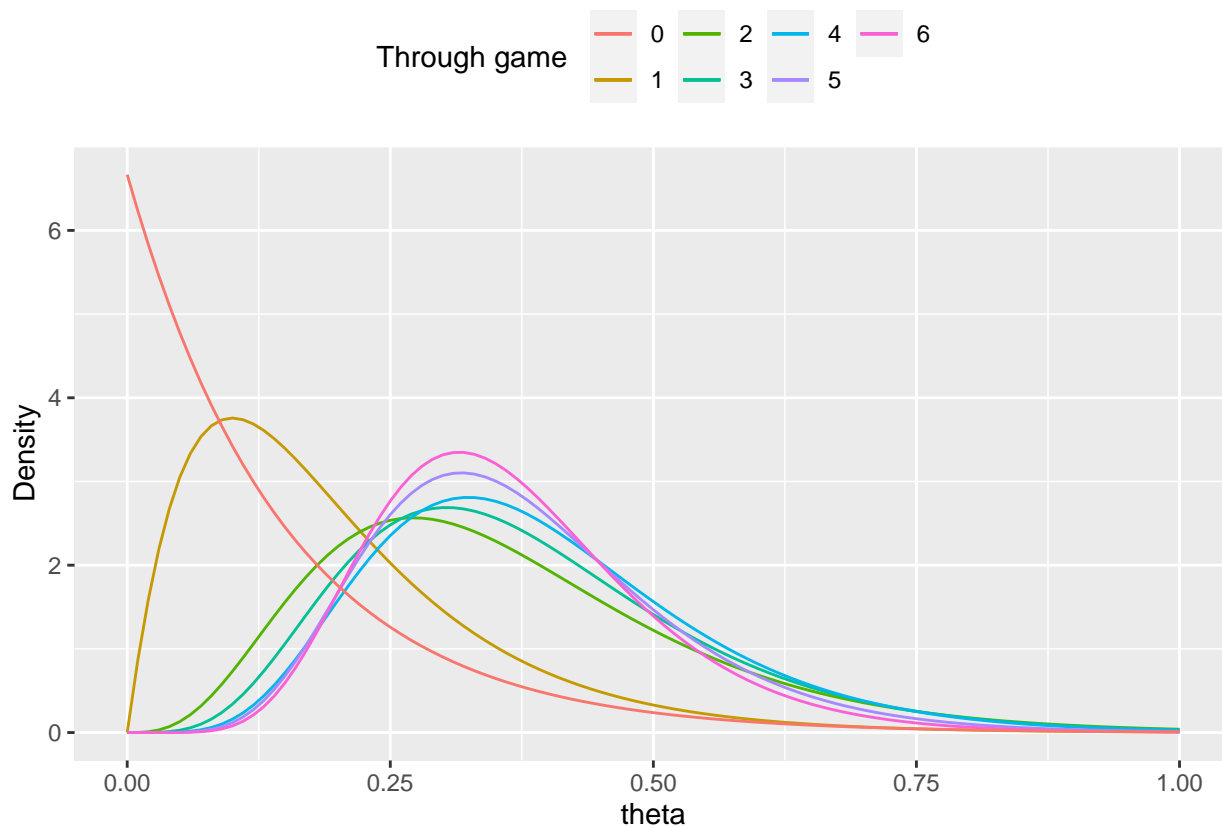
numerator_4 <- function(theta) {
  exp(dexp(theta, rate = 1 / m, log = TRUE) +
    log_likelihood(theta, game = game_1) +
    log_likelihood(theta, game = game_2) +
    log_likelihood(theta, game = game_3) +
    log_likelihood(theta, game = game_4))
}
denominator_4 <- integrate(numerator_4, lower = 0, upper = Inf)$value

numerator_5 <- function(theta) {
  exp(dexp(theta, rate = 1 / m, log = TRUE) +
    log_likelihood(theta, game = game_1) +
    log_likelihood(theta, game = game_2) +
    log_likelihood(theta, game = game_3) +
    log_likelihood(theta, game = game_4) +
    log_likelihood(theta, game = game_5))
}
denominator_5 <- integrate(numerator_5, lower = 0, upper = Inf)$value

numerator_6 <- function(theta) {
  exp(dexp(theta, rate = 1 / m, log = TRUE) +
    log_likelihood(theta, game = game_1) +
    log_likelihood(theta, game = game_2) +
    log_likelihood(theta, game = game_3) +
    log_likelihood(theta, game = game_4) +
    log_likelihood(theta, game = game_5) +
    log_likelihood(theta, game = game_6))
}
denominator_6 <- integrate(numerator_6, lower = 0, upper = Inf)$value

ggplot() + xlim(0, 1) + ylab("Density") + xlab("theta") +
  scale_color_discrete(name = "Through game ") + theme(legend.position = "top") +
  geom_function(fun = ~numerator_1(.x) / denominator_1, aes(color = "1")) +
  geom_function(fun = ~numerator_2(.x) / denominator_2, aes(color = "2")) +
  geom_function(fun = ~numerator_3(.x) / denominator_3, aes(color = "3")) +
  geom_function(fun = ~numerator_4(.x) / denominator_4, aes(color = "4")) +
  geom_function(fun = ~numerator_5(.x) / denominator_5, aes(color = "5")) +
  geom_function(fun = ~numerator_6(.x) / denominator_6, aes(color = "6")) +
  geom_function(fun = dexp, args = list(rate = 1 / m), aes(color = "0"))

```



The exponential prior with expectation  $m$  is given by the line “through game 0”. After the first game, the posterior density shifts away from near-zero values of  $\theta$ . The posterior density shifts right a bit more after the second game, and then becomes increasingly concentrated around  $\theta \approx \frac{1}{3}$ . Thus, our updated beliefs after having observed a few games are that this bowler is worse than we originally believed.

## 2.5 Simulate Score Distribution

Although it is plausible to assume that pins knocked down are independent across frames of bowling, the scoring rules for bowling are not that simple. The “traditional” scoring rules are described at

[https://en.wikipedia.org/wiki/Ten-pin\\_bowling#Traditional\\_scoring](https://en.wikipedia.org/wiki/Ten-pin_bowling#Traditional_scoring)

but basically they say that if you get a strike on frame  $j$ , then your score for frame  $j$  is 10 plus the number of pins knocked down on your next two rolls (for a maximum of 30). And if you get a spare on frame  $j$ , then your score for frame  $j$  is 10 plus the number of pins knocked down on the first roll of frame  $j + 1$  (for a maximum of 20). The tenth frame is scored as described above, and the additional rolls make the scoring consistent with the previous nine frames. A perfect game would achieve the maximum possible score of 300 with twelve consecutive strikes.

Write a R function called `score` that takes a matrix with 10 rows and 3 columns like that above that indicates how many pins were knocked down on that frame and that roll and returns the bowler’s score at the end of the game according to “traditional” scoring rules.

```
score <- function(game) {
  total <- sum(game, na.rm = TRUE)
  for (frame in 1:8) {
    if (game[frame, 1] == 10) { # strike
      total <- total + game[frame + 1, 1]
```

```

    if (game[frame + 1, 1] == 10) {
      total <- total + game[frame + 2, 1]
    } else {
      total <- total + game[frame + 1, 2]
    }
  } else if (sum(game[frame, 1:2]) == 10) { # spare
    total <- total + game[frame + 1, 1]
  }
}
if (game[9, 1] == 10) { # strike on frame 9
  total <- total + game[10, 1] + game[10, 2]
} else if (sum(game[9, 1:2]) == 10) { # spare on frame 9
  total <- total + game[10, 1]
}
total
}

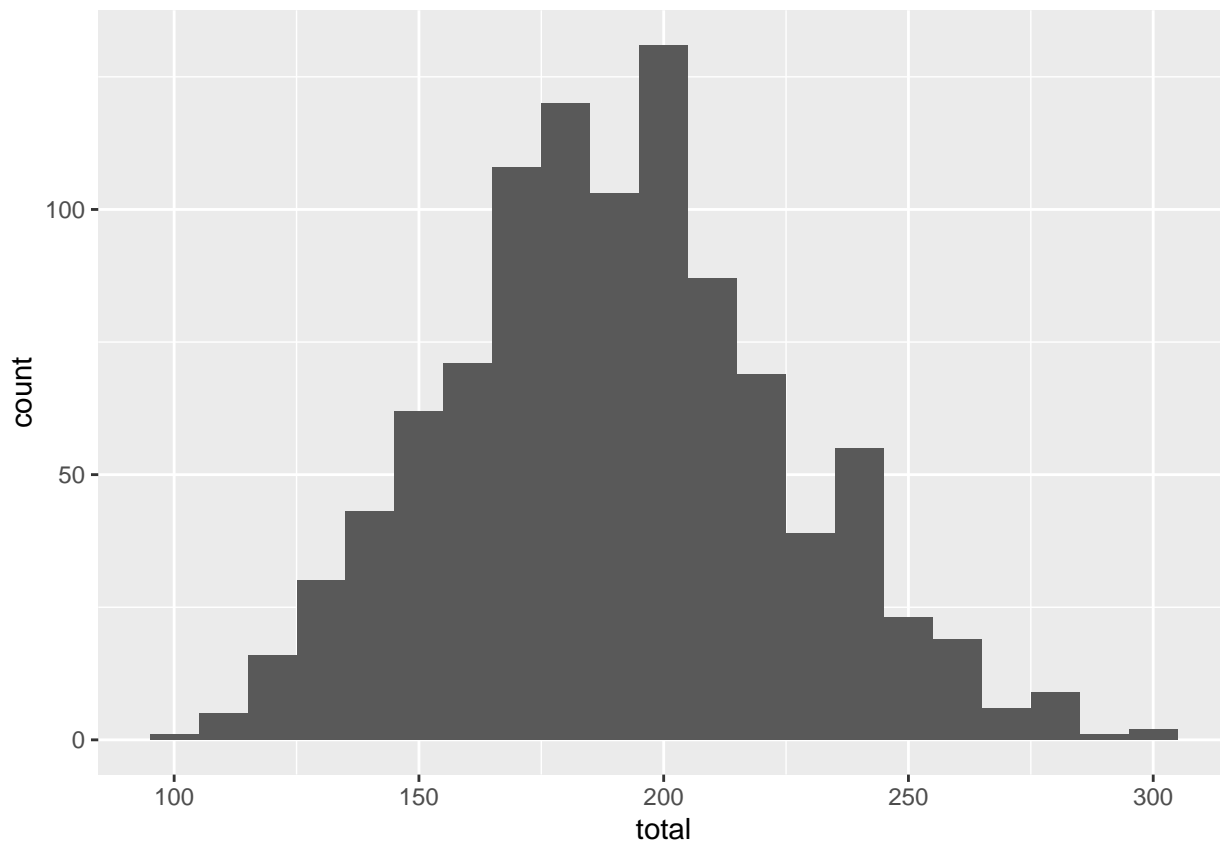
```

Then, take 1000 draws from your exponential prior (not posterior) for  $\theta$ . For each of those 1000 realizations of  $\theta$ , simulate one entire game of bowling (represented by an  $11 \times 3$  matrix), and pass that matrix to your score function. In the end, you should have 1000 integers between 0 and 300. Create a histogram out of these 1000 integers. What is the mode of the bowler's score under your prior beliefs about  $\theta$ ?

```

sim <- tibble(theta = rexp(1000, rate = 1 / m),
  total = sapply(theta, FUN = function(t) {
    game <- matrix(NA_integer_, nrow = 11, ncol = 3)
    for (frame in 1:9) {
      x_1 <- sample(Omega, size = 1, prob = Pr(Omega, n = 10, theta = t))
      game[frame, 1] <- x_1
      if (x_1 < 10) {
        x_2 <- sample(Omega, size = 1, prob = Pr(Omega, n = 10 - x_1, theta = t))
        game[frame, 2] <- x_2
      }
    }
    x_1 <- sample(Omega, size = 1, prob = Pr(Omega, n = 10, theta = t))
    game[10, 1] <- x_1
    if (x_1 == 10) {
      x_2 <- sample(Omega, size = 1, prob = Pr(Omega, n = 10, theta = t))
      game[10, 2] <- x_2
      if (x_2 == 10) {
        x_3 <- sample(Omega, size = 1, prob = Pr(Omega, n = 10, theta = t))
      } else {
        x_3 <- sample(Omega, size = 1, prob = Pr(Omega, n = 10 - x_2, theta = t))
      }
      game[10, 3] <- x_3
    } else {
      x_2 <- sample(Omega, size = 1, prob = Pr(Omega, n = 10 - x_1, theta = t))
      game[10, 2] <- x_2
    }
    score(game)
  })
ggplot(sim) + geom_histogram(aes(x = total), binwidth = 10)

```



```
count(sim, total) %>%
  arrange(desc(n)) %>%
  slice_head(n = 1)
```

```
## # A tibble: 1 x 2
##   total     n
##   <int> <int>
## 1   166    19
```

which is the mode. Note that the mode is very fragile, in the sense that it could easily be some other number for 1000 simulations, and is not that close to the median or mean.