

Section 4: Intervals, PPCS, and Rejection Sampling

Hot Hand

The “hot hand” is the purported phenomenon that a person who experiences a successful outcome has a greater chance of success in further attempts. The concept originates from basketball whereas a shooter is allegedly more likely to score if their previous attempts were successful. While previous success at a task can indeed change the psychological attitude and subsequent success rate of a player, researchers for many years did not find evidence for a “hot hand” in practice, dismissing it as fallacious. However, later research questioned whether the belief is indeed a fallacy. In this lab, we will boil down this question as testing whether the results of shots are independent.

Let “1” denote a valid shot and “0” denote an invalid. Suppose we observe the following results of a player:

```
# observations #
set.seed(123)
y <- c(rep(1, 18), rep(0, 3), rep(1, 6), rep(0, 2),
      rbinom(67, 1, prob = 0.25), rep(1, 4))
```

Suppose $Y_i \sim \text{Bernoulli}(\theta)$ and $\theta \sim \text{Beta}(3, 7)$

Find the posterior:

```
# prior #
a <- 3
b <- 7
#posterior #
a_post <- a + sum(y)
b_post <- b + (length(y) - sum(y))
a_post; b_post
```

```
## [1] 50
```

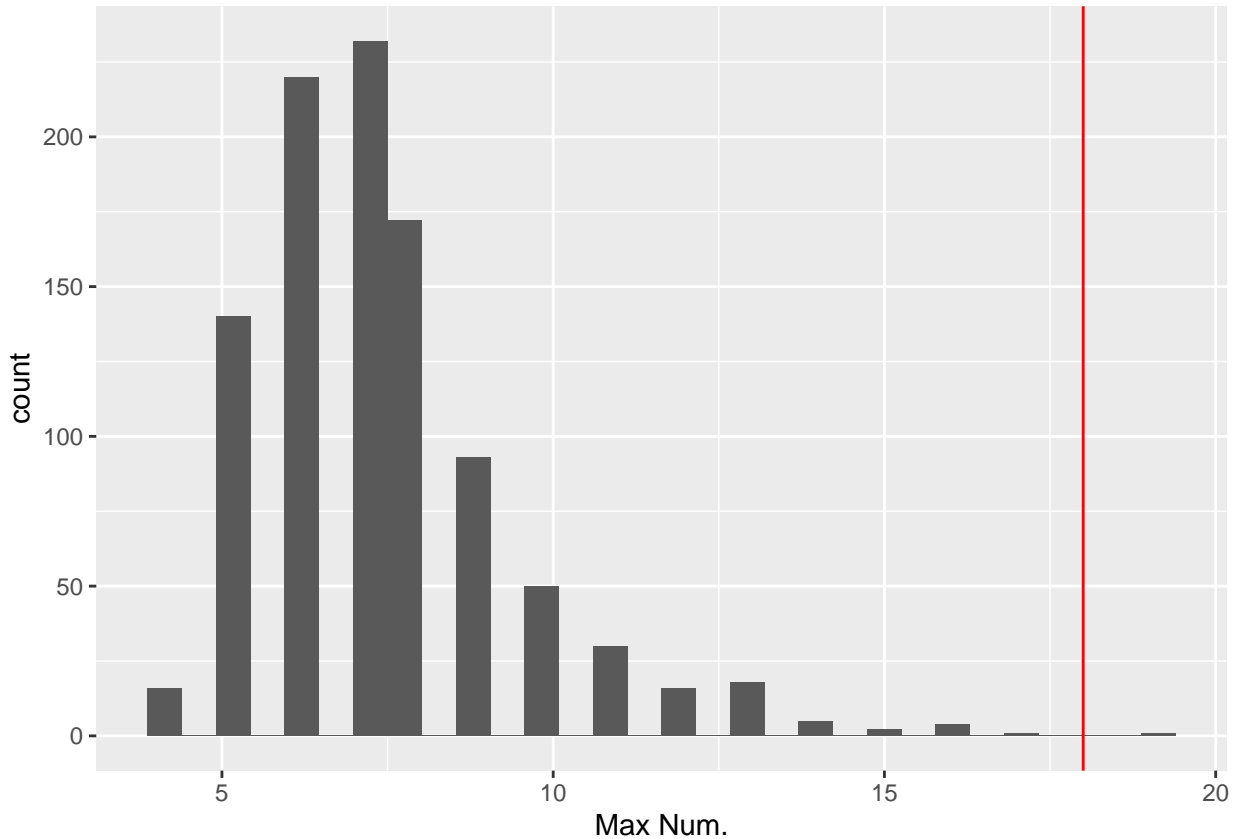
```
## [1] 60
```

Let the test stat. be the maximum number of the same consecutive results. (Other good test stat. might be the fraction of times you see (1, 1) vs (1, 0).)

```
# observed test stat. #
test_stat_obs <- max(rle(y)$lengths)

# test stat. based on simulation #
nsim <- 1000
test_stat_rep <- rep(NA, nsim)
for (i in 1:1000) {
  p_post <- rbeta(1, a_post, b_post)
  y_rep <- rbinom(100, size = 1, prob = p_post)
  test_stat <- max(rle(y_rep)$lengths)
  test_stat_rep[i] <- test_stat
}

ggplot(tibble(test_stat_rep), aes(test_stat_rep)) +
  geom_histogram() + xlab("Max Num.") +
  geom_vline(xintercept = test_stat_obs, colour = "red")
```



In this example, the data doesn't look independent. The “hot-hand” phenomenon might exist.

In this section we'll review rejection sampling.

Rejection Sampling

Rejection sampling is easiest to think of graphically. We first plot the probability density function or probability mass function of the distribution that we want to sample from. If we can uniformly sample from a region that bounds this pdf/pmf, then we can do rejection sampling by:

1. Rejection Sampling for the Semi-Circle Distribution. Let

$$p(x|R) = \frac{2}{\pi R^2} \sqrt{R^2 - x^2} \mathbf{1}\{x \in [-R, R]\}$$

- (a) Identify a proposal distribution, $g(x | R)$ for rejection sampling. Show that there exists some M such that $Mg(x | R) \geq p(x|R) \forall x \in [-R, R]$. What is M ?

proposal distribution:

$$g(x|R) = \frac{1}{2R} \mathbf{1}\{x \in [-R, R]\}$$

$$\begin{aligned} \frac{p(x|R)}{g(x|R)} &= \frac{4}{\pi R} \sqrt{R^2 - x^2} \\ &\leq \frac{4}{\pi R} R \\ &= \frac{4}{\pi} \end{aligned}$$

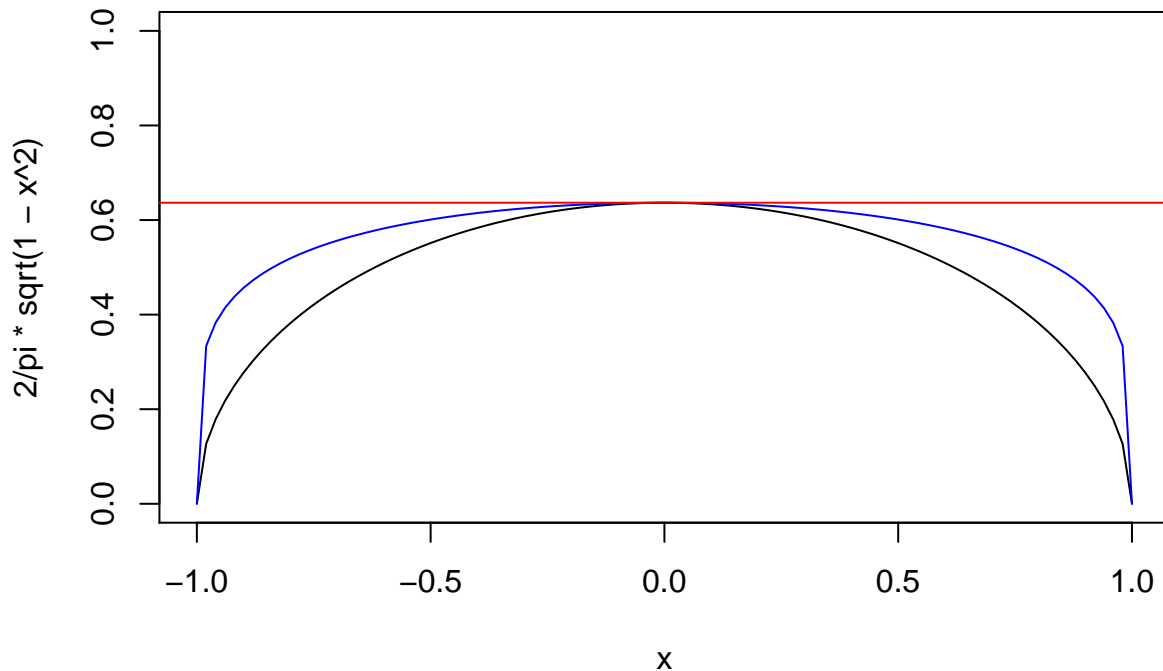
Therefore, M can choose $\frac{4}{\pi}$.

- (a) Plot the semi-circle distribution for $R = 1$ and your proposal distribution on the same plot using the `curve` function. Note that because the density must be normalized to integrate to 1, the semi-circle distribution is really a “semi-ellipse”.

```
ratio_densities <- function(x) {
  (2/pi*sqrt(1-x^2)) / (2*dbeta((x + 1)/2, 1.2, 1.2))
}

M <- optimize(ratio_densities, lower=-1, upper=1, maximum=TRUE)$objective

curve(2/pi*sqrt(1-x^2), from=-1, 1, ylim=c(0, 1))
curve(M*2*dbeta((x+1)/2, 1.2, 1.2), add=TRUE, col="blue")
abline(h=2/pi, col="red")
```



#. Use rejection sampling to sample from the semi-circle distribution with $R=1$. What is the acceptance rate?

```
N = 10000
R = 1
M = 4/pi
x <- runif(N, -R, R)
y <- runif(N, 0, 1)
p <- function(x, R) {
  return(2/pi/R^2 * sqrt(R^2 - x^2))
}
g <- function(x, R) {
  return(1/2/R)
}
decision <- y <= sapply(x, p, R = 1)/M /sapply(x, g, R = 1)

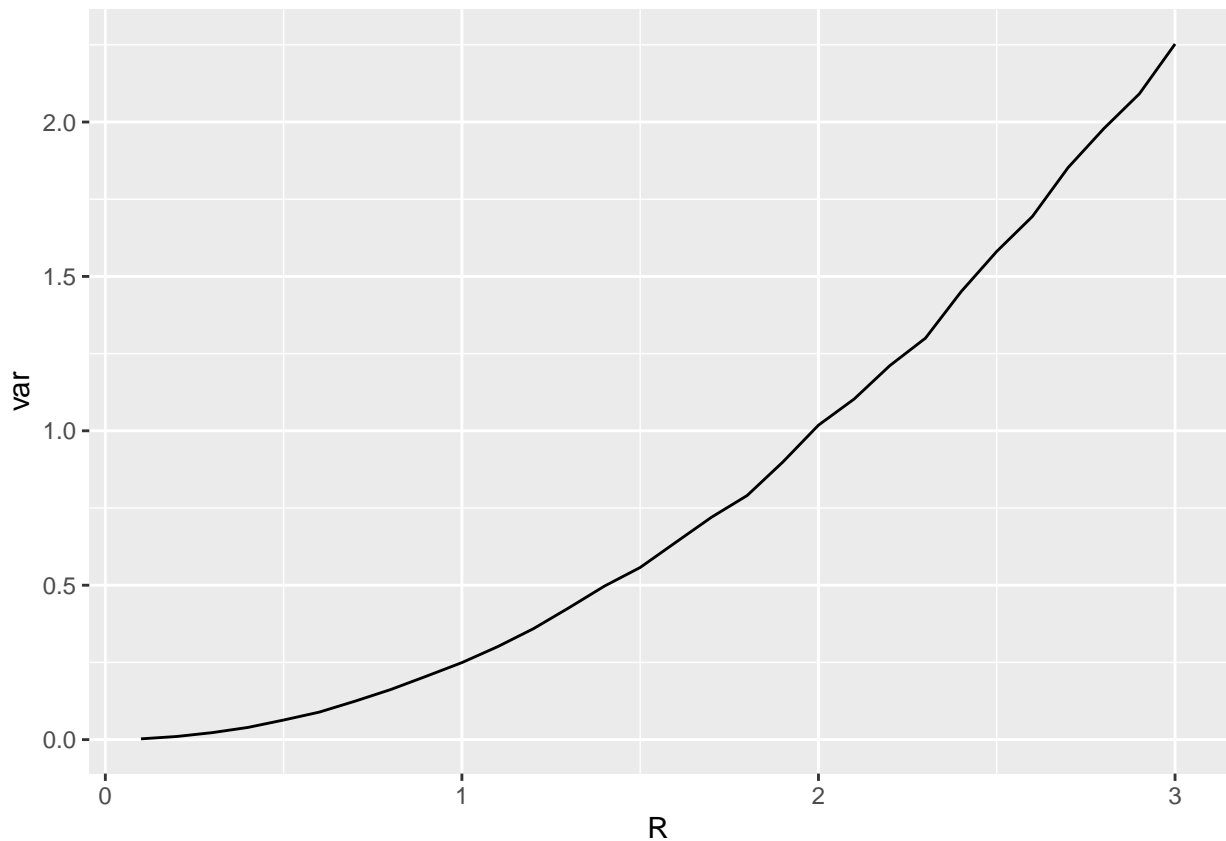
x = x[decision]
sum(decision)/ N
```

```
## [1] 0.7802
```

#. For $R = 0.1, 0.2, \dots, 3.0$, use rejection sampling to sample from $p(x|R)$. Plot the sample variance for each

value of R , and show graphically that the variance for the semi-circle distribution is $\frac{R^2}{4}$.

```
R = seq(0.1, 3.0, by = 0.1)
M = 4/pi
N = 10000
v <- rep(0, length(R))
for (i in 1 : length(R)) {
  r = R[i]
  x <- runif(N, -r, r)
  y <- runif(N, 0, 1)
  decision <- y <= sapply(x, p, R = r)/M /sapply(x, g, R = r)
  x = x[decision]
  v[i] = var(x)
}
v <- v %>% as.data.frame() %>% cbind(R)
colnames(v) <- c("var", "R")
ggplot(v, aes(x = R, y = var)) + geom_line()
```



can be checked that the line resembles $var = R^2/4$.

```
N = 1000

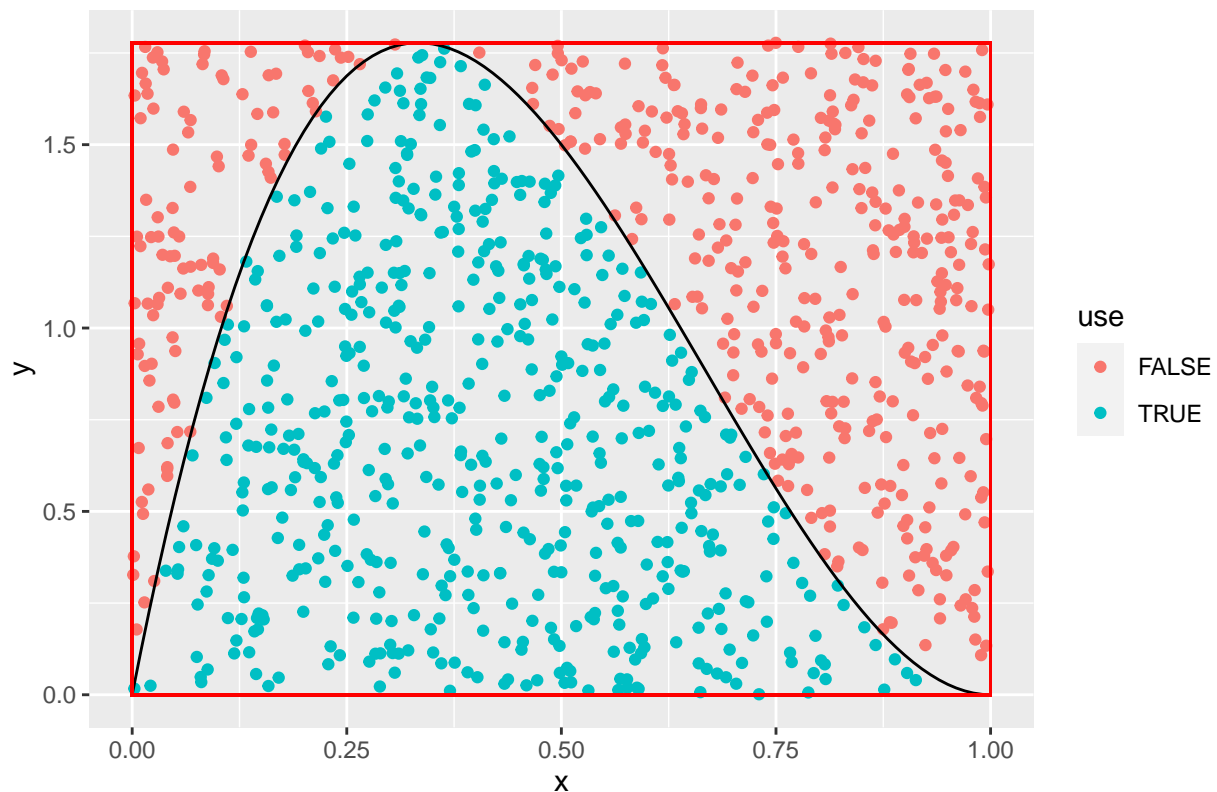
alpha = 2.0
beta = 3.0
mode = (alpha - 1) / (alpha + beta - 2)
maxB = dbeta(mode, alpha, beta)

list(x = runif(N),
     y = runif(N) * maxB) %>%
```

```
as.tibble %>%
mutate(use = y < dbeta(x, alpha, beta)) %>%
ggplot(aes(x, y)) +
geom_point(aes(color = use)) +
stat_function(fun = function(x) dbeta(x, alpha, beta)) +
geom_rect(aes(xmin = 0.0, xmax = 1.0, ymin = 0.0, ymax = maxB), alpha = 0.0, color = "red") +
ggtitle("Proposal distribution outlined in red, target in black")
```

```
## Warning: `as.tibble()` is deprecated as of tibble 2.0.0.
## Please use `as_tibble()` instead.
## The signature and semantics have changed, see `?as_tibble`.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

Proposal distribution outlined in red, target in black



The points highlighted in teal above have the desired distribution. We throw away the reddish samples.