

MATH 3070 Lab Project 9

Pranav Rajan

October 29, 2020

Contents

Problem 1 (Verzani problem 6.2)	1
Problem 2 (Verzani problem 6.3)	1
Problem 3 (Verzani problem 6.23)	2
Problem 4	2

*Remember: I expect to see commentary either in the text, in the code with comments created using #, or (preferably) both! **Failing to do so may result in lost points!***

Since this assignment involves simulation, I set the seed to the following in order to get the same results:

```
set.seed(5292016)
```

Problem 1 (Verzani problem 6.2)

Roll a pair of dice. Let X be the largest value shown on the two dice. Use `sample()` to simulate five values of X .

```
# Your code here
list <- c(1:5)
for (value in list) {
  d1 <- sample(1:6, size=1, replace = TRUE)
  d2 <- sample(1:6, size=1, replace=TRUE)
  print(max(d1, d2))
}
```

```
## [1] 6
## [1] 2
## [1] 6
## [1] 6
## [1] 3
```

Problem 2 (Verzani problem 6.3)

The National Basketball Association lottery to award the first pick in the draft is held by putting 1,000 balls into a hopper and selecting one. The teams with the worst records the previous year have a greater proportion of the balls. The data set `nba.draft` (*UsingR*) contains the ball allocation for the year 2002. Use `sample()`

with `Team` as the data vector and `prob=Balls` to simulate the draft. What team do you select? Repeat until Golden State is chosen. How long did it take?

```
# Your code here
suppressMessages(library(UsingR))
require(UsingR)
sample(nba.draft$Team, size = 1, replace=TRUE, prob = nba.draft$Balls)

## [1] Chicago
## 13 Levels: Atlanta Chicago Cleveland Denver Golden State ... Washington

# first run result: Golden State
# number of attempts to get to Golden State: 1
```

Problem 3 (Verzani problem 6.23)

Find the quintiles (0th, 20th, 40th, 60th, 80th, and 100th quantiles/percentiles) of the standard Normal distribution (this is a normal distribution with mean 0 and standard deviation 1).

```
# Your code here
qnorm(c(0.0, 0.2, 0.4, 0.6, 0.8, 1.0), mean = c(0, 0, 0, 0, 0, 0), sd = c(1, 1, 1, 1, 1, 1))

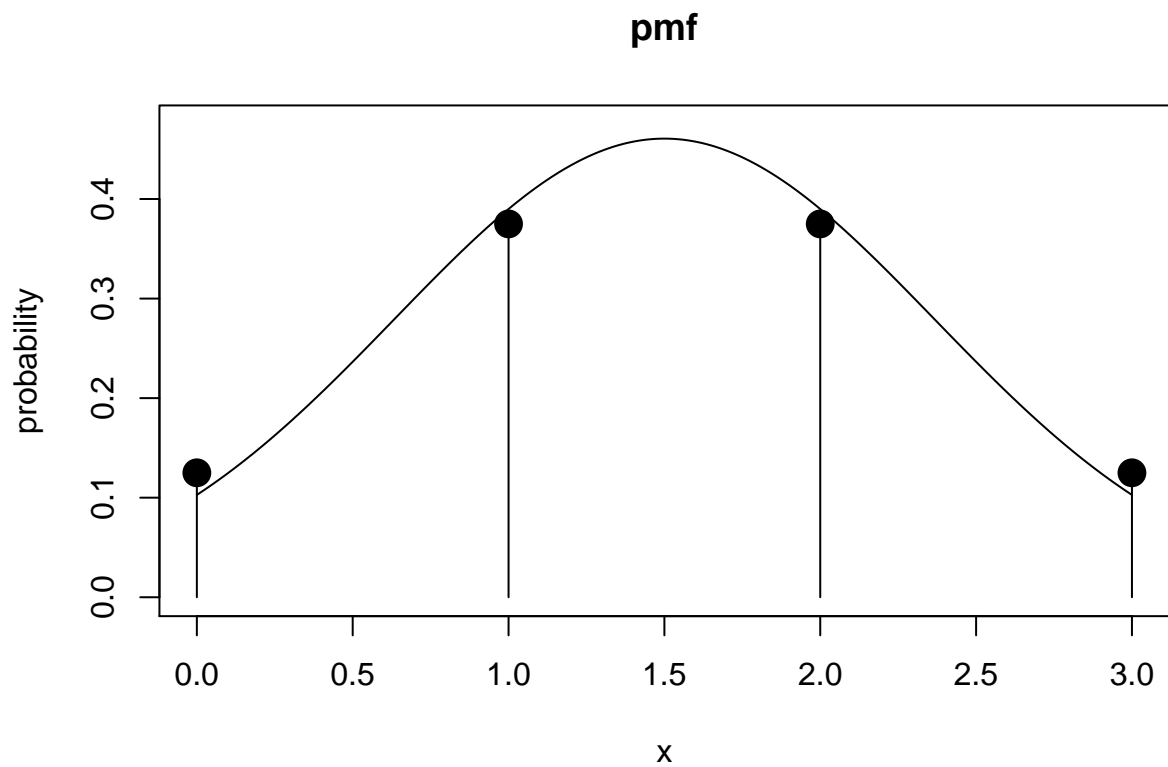
## [1] -Inf -0.8416212 -0.2533471 0.2533471 0.8416212 Inf
```

Problem 4

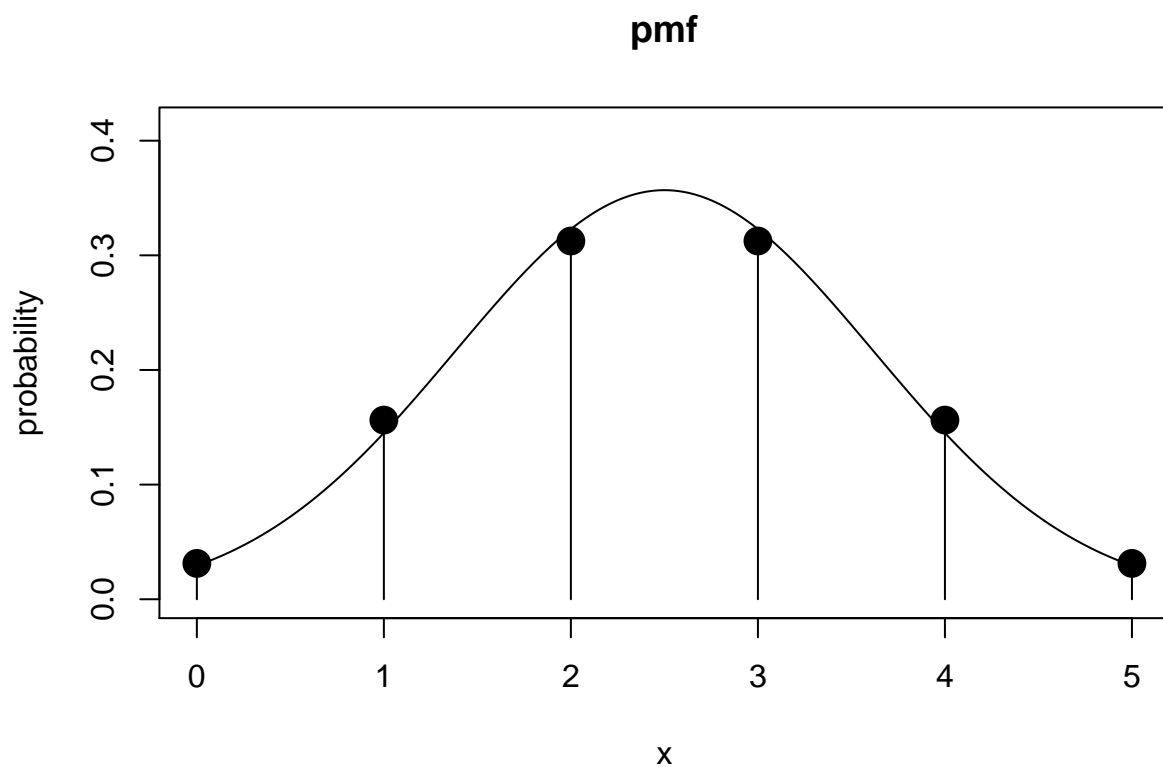
Consider flipping a fair coin n times and counting the number of time the coin lands heads-up. This is a binomial random variable, but it turns out that as n gets large, this distribution can be approximated with a Normal distribution, where the mean is $0.5 \times n$ and the standard deviation is $0.5 \times \sqrt{n}$. We know this thanks to the central limit theorem. When flipping 3, 5, and 20 times, plot the pmf of the binomial random variable with the appropriate parameters (you may use my function `plot_pmf()` from the lecture notes). Superimpose on each of these plots the density curve of the Normal distribution with the appropriate parameters (use the `lines()` function to do so; an example for doing this is in the lecture notes). What do you notice? When does the approximation appear best?

```
# Your code here
plot_pmf <- function(q, p) {
  # This will plot a series of horizontal lines at q with height p, setting
  # the y limits to a reasonable heights
  plot(q, p, type = "h", xlab = "x", ylab = "probability", main = "pmf", ylim = c(0,
    max(p) + 0.1))
  # Usually these plots have a dot at the end of the line; the point function
  # will add these dots to the plot created above
  points(q, p, pch = 16, cex = 2)
}

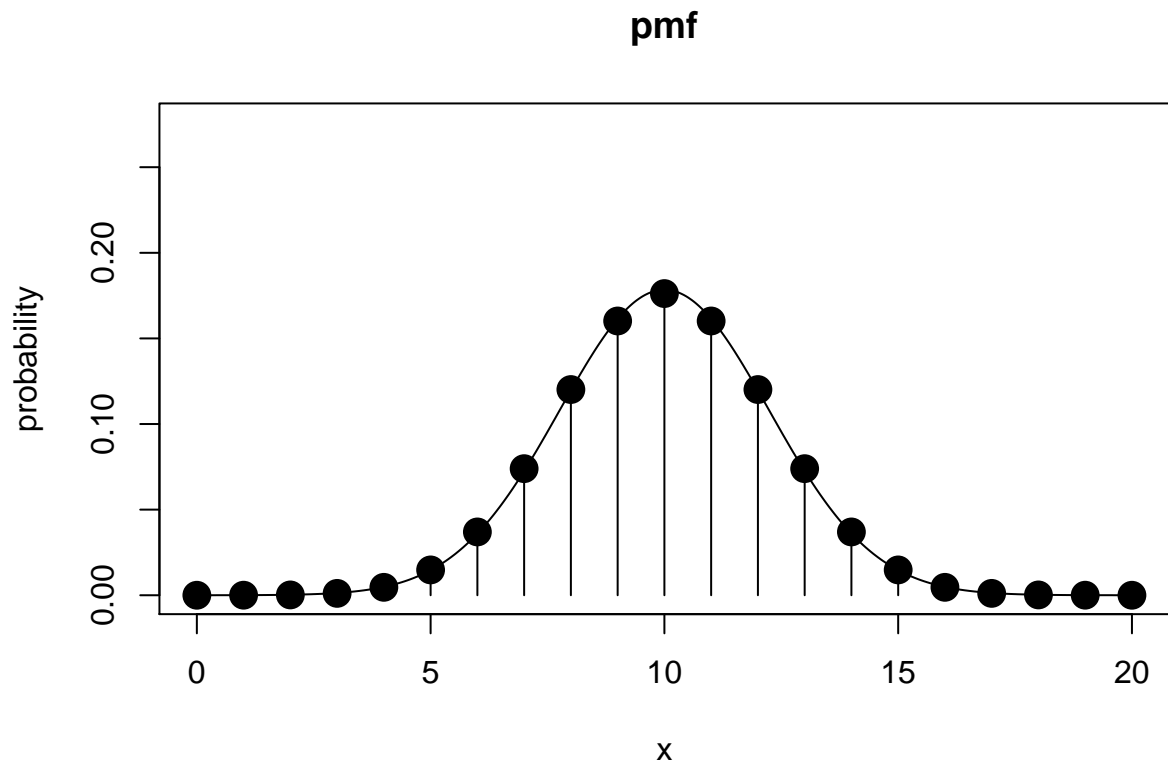
# 3 flips
binom_q3 <- 0:3
binom_pmf3 <- dbinom(binom_q3, size = 3, prob = 0.5)
clt_norm_q3 <- seq(0, 3, length = 1000)
plot_pmf(binom_q3, binom_pmf3)
lines(clt_norm_q3, dnorm(clt_norm_q3, mean = 0.5 * 3, sd = 0.5 * sqrt(3)), col = "black")
```



```
# 5 flips
binom_q5 <- 0:5
binom_pmf5 <- dbinom(binom_q5, size = 5, prob = 0.5)
clt_norm_q5 <- seq(0, 5, length = 1000)
plot_pmf(binom_q5, binom_pmf5)
lines(clt_norm_q5, dnorm(clt_norm_q5, mean = 0.5 * 5, sd = 0.5 * sqrt(5)), col = "black")
```



```
# 20 flips
binom_q20 <- 0:20
binom_pmf20 <- dbinom(binom_q20, size = 20, prob = 0.5)
clt_norm_q20 <- seq(0, 20, length = 1000)
plot_pmf(binom_q20, binom_pmf20)
lines(clt_norm_q20, dnorm(clt_norm_q20, mean = 0.5 * 20, sd = 0.5 * sqrt(20)), col = "black")
```



```
# Explanation
# The more flips you have the density curve starts looking closer like the normal distribution.
# The approximation looks best when we have 20 flips. The reason is because we have a lot more
# points which allows us to visualize the distribution better.
```