MATH 3070 Lab Project 10

Prachi Aswani

November 7, 2024

Contents

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.*

```
# Simulating the dice rolls
# We'll roll two dice five times each and store the results
die1_rolls <- sample(1:6, 5, replace = TRUE)  # Roll the first die 5 times
die2_rolls <- sample(1:6, 5, replace = TRUE)  # Roll the second die 5 times

# Calculating X as the maximum value between each pair of dice rolls
X_values <- pmax(die1_rolls, die2_rolls)  # pmax finds the element-wise maximum

# Display the five simulated values of X
X_values
```

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

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?*

```r
# Load the data
# Loading libraries
library(UsingR)
```

```
## Warning: package 'UsingR' was built under R version 4.3.3

## Loading required package: MASS

## Loading required package: HistData

## Warning: package 'HistData' was built under R version 4.3.3

## Loading required package: Hmisc

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```r
data("nba.draft")

# Initialize counter for number of draws
draw_count <- 0
selected_team <- ""

# Loop until Golden State is chosen
while (selected_team != "Golden State") {
  # Simulate a draft draw based on team probabilities
  selected_team <- sample(nba.draft$Team, 1, prob = nba.draft$Balls)

  # Increment draw count
  draw_count <- draw_count + 1
}

# Output the result
draw_count  # Number of draws until Golden State was chosen
```

```
## [1] 5
```

```r
selected_team  # Should display "Golden State"
```

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

```r
# The first team selected was "Atlanta"
# It took 5 draws to finally select "Golden State."
```

Problem 3 (Verzani problem 6.23)

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

```r
# Define the quantile levels
quartiles <- c(0, 0.2, 0.4, 0.6, 0.8, 1)

# Calculate the corresponding values of the standard normal distribution
quintile_values <- qnorm(quartiles, mean = 0, sd = 1)

# Display the results
quintile_values
```

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

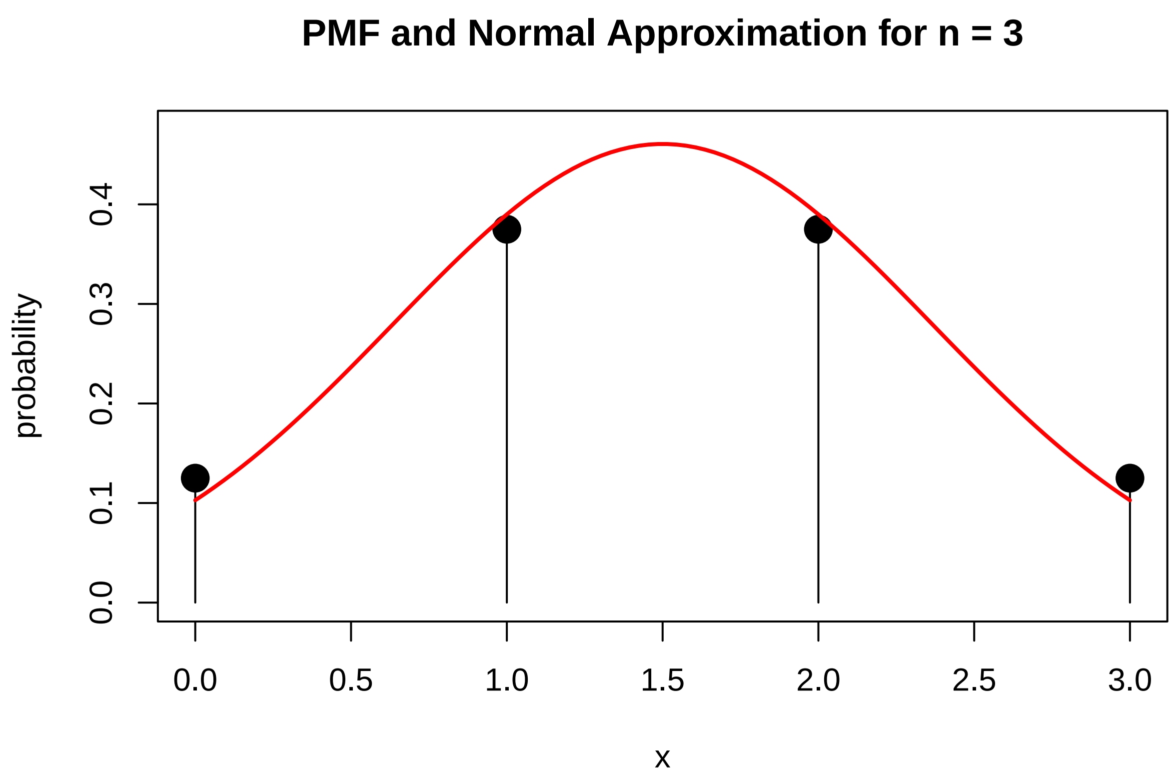
Problem 4

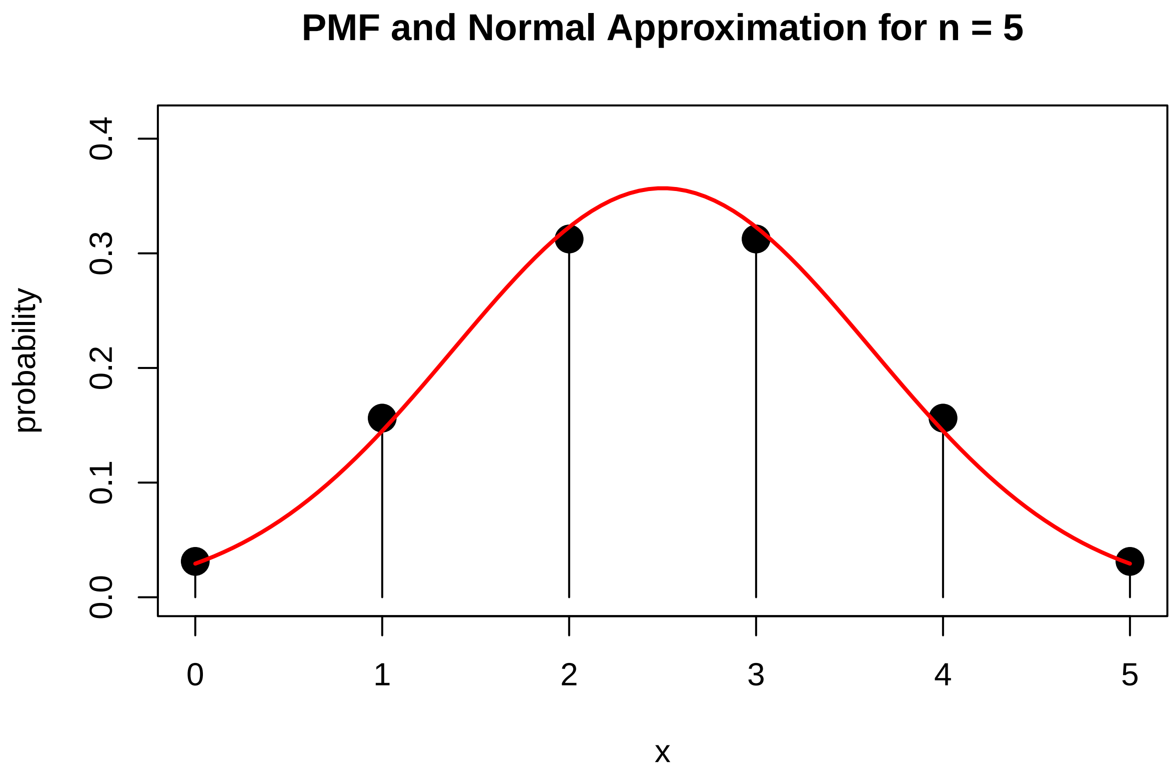
Consider flipping a fair coin n times and counting the number of time the coin lands heads-up. This is a binomial random variable, but it turns out that as n gets large, this distribution can be approximated with a Normal distribution, where the mean is $0.5 \times n$ and the standard deviation is $0.5 \times \sqrt{n}$. 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?

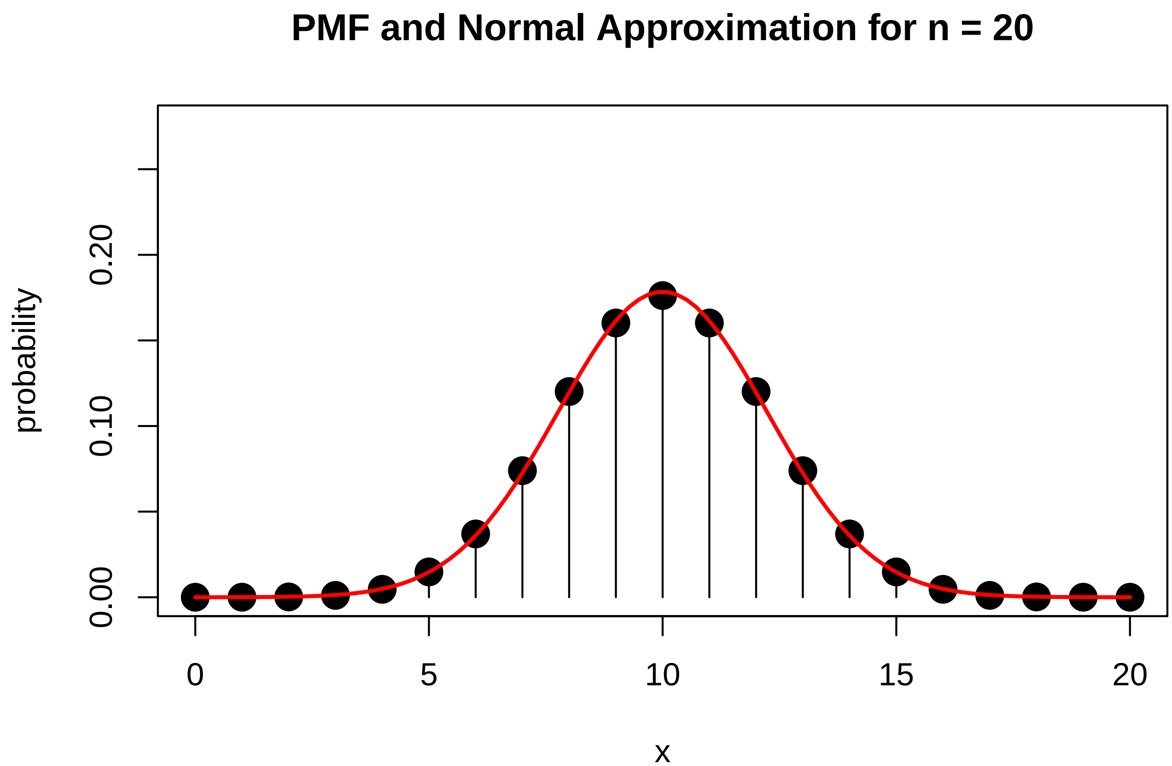
```r
plot_pmf <- function(q, p) {
    # This will plot a series of horizontal lines at q with height p, setting
    # the y limits to a reasonable heights
    plot(q, p, type = "h", xlab = "x", ylab = "probability", ylim = c(0,
        max(p) + 0.1))
    # Usually these plots have a dot at the end of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

```r
# Your code here
# Define the values of n for different scenarios
n_values <- c(3, 5, 20)

# Loop through each n value and create the PMF and normal approximation
for (n in n_values) {
    # Parameters for the binomial distribution
    p <- 0.5
    x_vals <- 0:n  # Possible values for the number of heads
    pmf_binomial <- dbinom(x_vals, size = n, prob = p)

    # Plot the PMF using the plot_pmf function
    plot_pmf(x_vals, pmf_binomial)
    title(main = paste("PMF and Normal Approximation for n =", n))
```

```r
    # Parameters for the normal distribution
    mean_normal <- 0.5 * n
    sd_normal <- 0.5 * sqrt(n)

    # Generate x values for the normal curve (from slightly below 0 to slightly above n)
    x_norm <- seq(0, n, length.out = 100)
    density_normal <- dnorm(x_norm, mean = mean_normal, sd = sd_normal)

    # Superimpose the normal distribution density on the PMF plot
    lines(x_norm, density_normal, col = "red", lwd = 2)
}
```




PMF and Normal Approximation for n = 5
probability
0.0
0.1
0.2
0.3
0.4
0
1
2
3
4
5
x




```
# For small values of n (like 3 and 5), the binomial distribution is discrete
# and the normal approximation is not a very good fit. The PMF appears more
# "spiky," and the normal curve doesn't align well with the data.

# As n increases (in the case of n=20), the normal distribution starts to fit the
# binomial distribution much better. The PMF becomes smoother, and the red normal
# curve closely matches the shape of the binomial distribution.

# The normal approximation seems to work best when n is large, as shown in the n=20 plot.
# The binomial distribution approaches a bell-shaped curve, making the normal
# approximation more accurate. The approximation improves as n increases because
# of the central limit theorem, which states that the distribution of the sum of
# a large number of independent and identically distributed random variables will
# tend toward a normal distribution.
```