Lab4

Exercises

Irimie Fabio

Contents

Exercise 1

A

Plot the Probability Mass Function for the Binomial distribution with $n = 18$ and $p = \frac{1}{3}$. Calculate:

1. $P(X = 3)$

```
dbinom(3, 18, 1 / 3)
## [1] 0.06901723
```

2. $P(X \geq 3)$

```
1 - pbinom(2, 18, 1 / 3)
## [1] 0.9673521
# or
```

```
pbinom(2, 18, 1 / 3, lower.tail = FALSE)
## [1] 0.9673521
```

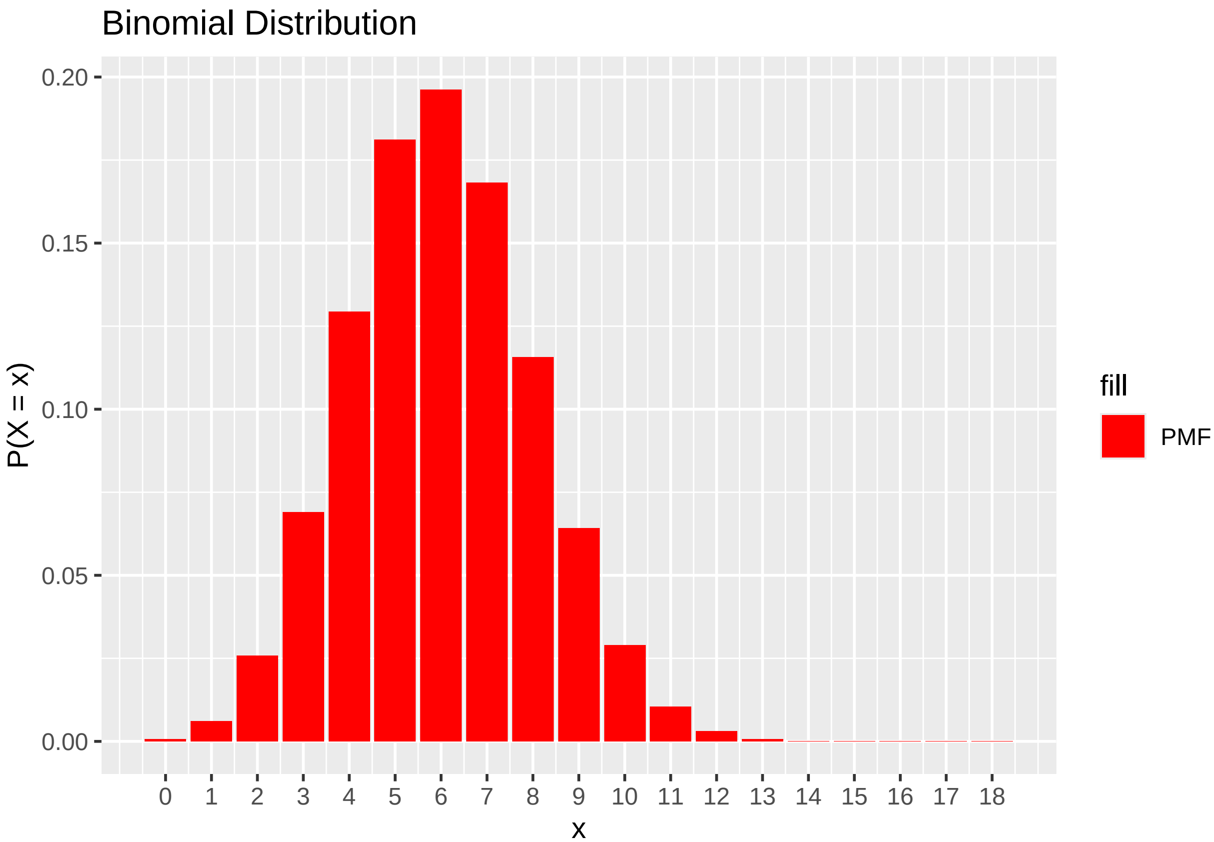
3. $P(1 \leq X < 5)$

```
pbinom(4, 18, 1 / 3) - pbinom(0, 18, 1 / 3)
## [1] 0.2303957
```

4. $P(X \geq 15)$

```
pbinom(14, 18, 1 / 3, lower.tail = FALSE)
## [1] 1.852509e-05
```

```
library(ggplot2)
df <- data.frame(x = 0:18, y = dbinom(0:18, 18, 1 / 3))

ggplot(df, aes(x = x, y = y, fill = "PMF")) +
  geom_col() +
  scale_x_continuous(breaks = 0:18) +
  scale_fill_manual(values = "red") +
  labs(
    title = "Binomial Distribution",
    x = "x",
    y = "P(X = x)"
  )
```



B

Plot the Cumulative Distribution Function for the Poisson distribution with $\lambda = 3$. Calculate:

1. $P(X = 3)$

```
lambda <- 3
dpois(3, lambda)
## [1] 0.2240418
```

2. $P(X \geq 3)$

```
ppois(2, lambda, lower.tail = FALSE)
## [1] 0.5768099
```

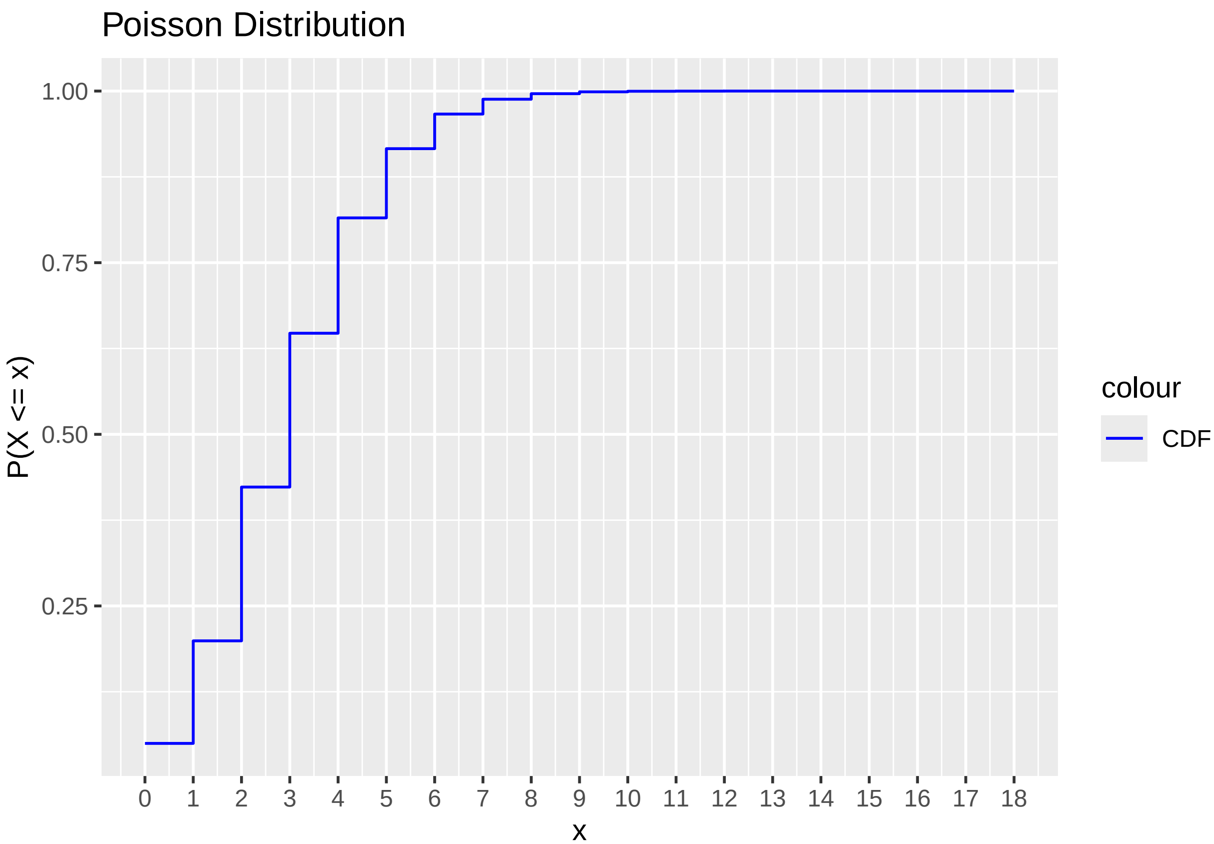
3. $P(1 \leq X > 5)$

```
ppois(4, lambda) - ppois(0, lambda)
## [1] 0.7654762
```

4. $P(X \geq 15)$

```
ppois(14, lambda, lower.tail = FALSE)
## [1] 6.703859e-07
```

```
df <- data.frame(x = 0:18, y = ppois(0:18, lambda))

ggplot(df, aes(x = x, y = y, color = "CDF")) +
  geom_step() +
  scale_x_continuous(breaks = 0:18) +
  scale_color_manual(values = "blue") +
  labs(
    title = "Poisson Distribution",
    x = "x",
    y = "P(X <= x)"
  )
```



Exercise 2

Demonstrate that a Poisson r.v. may be used as an approximation for a binomial r.v.

A

```
n <- c(20, 30, 40, 100)
p <- c(1 / 4, 1 / 6, 1 / 8, 1 / 20)

pmf <- matrix(NA, nrow = 21, ncol = 4)

for (i in 1:4) {
  pmf[, i] <- dbinom(0:20, n[i], p[i])
}

pmf <- as.data.frame(pmf)

colnames(pmf) <- paste("Binomial", n, round(p, 2), sep = "_")

pmf$Poisson <- dpois(0:20, n * p)

pmf$X <- 0:20
```

B

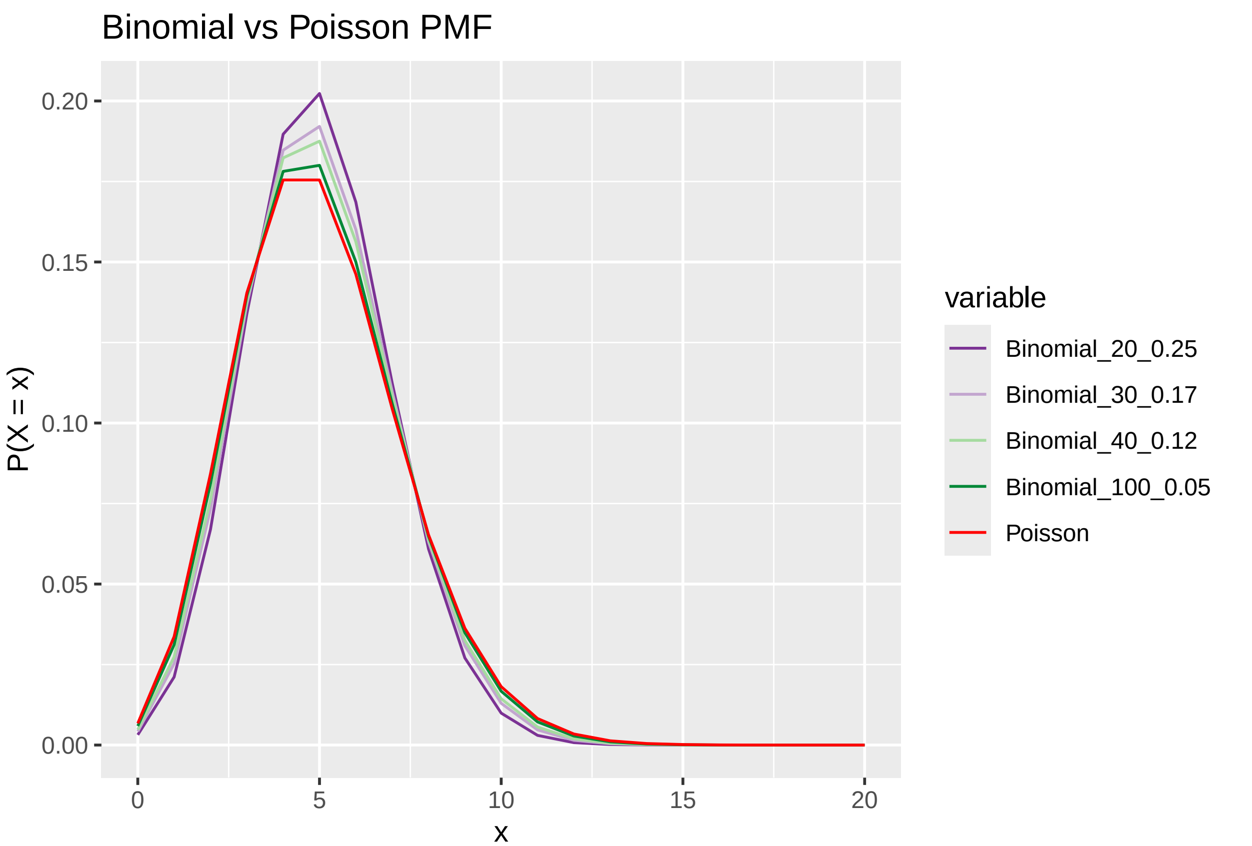
```
library(reshape2)

df_plot <- melt(pmf, id.vars = "X")
df_plot
```

```
##       X          variable        value
## 1     0  Binomial_20_0.25 3.171212e-03
## 2     1  Binomial_20_0.25 2.114141e-02
## 3     2  Binomial_20_0.25 6.694781e-02
## 4     3  Binomial_20_0.25 1.338956e-01
## 5     4  Binomial_20_0.25 1.896855e-01
## 6     5  Binomial_20_0.25 2.023312e-01
## 7     6  Binomial_20_0.25 1.686093e-01
## 8     7  Binomial_20_0.25 1.124062e-01
## 9     8  Binomial_20_0.25 6.088669e-02
## 10    9  Binomial_20_0.25 2.706075e-02
## 11   10  Binomial_20_0.25 9.922275e-03
## 12   11  Binomial_20_0.25 3.006750e-03
## 13   12  Binomial_20_0.25 7.516875e-04
## 14   13  Binomial_20_0.25 1.541923e-04
## 15   14  Binomial_20_0.25 2.569872e-05
## 16   15  Binomial_20_0.25 3.426496e-06
## 17   16  Binomial_20_0.25 3.569266e-07
## 18   17  Binomial_20_0.25 2.799425e-08
## 19   18  Binomial_20_0.25 1.555236e-09
## 20   19  Binomial_20_0.25 5.456968e-11
## 21   20  Binomial_20_0.25 9.094947e-13
## 22    0  Binomial_30_0.17 4.212720e-03
## 23    1  Binomial_30_0.17 2.527632e-02
## 24    2  Binomial_30_0.17 7.330133e-02
## 25    3  Binomial_30_0.17 1.368292e-01
## 26    4  Binomial_30_0.17 1.847194e-01
## 27    5  Binomial_30_0.17 1.921081e-01
## 28    6  Binomial_30_0.17 1.600901e-01
## 29    7  Binomial_30_0.17 1.097761e-01
## 30    8  Binomial_30_0.17 6.312124e-02
## 31    9  Binomial_30_0.17 3.085927e-02
## 32   10  Binomial_30_0.17 1.296090e-02
## 33   11  Binomial_30_0.17 4.713053e-03
## 34   12  Binomial_30_0.17 1.492467e-03
## 35   13  Binomial_30_0.17 4.132985e-04
## 36   14  Binomial_30_0.17 1.003725e-04
## 37   15  Binomial_30_0.17 2.141280e-05
## 38   16  Binomial_30_0.17 4.014899e-06
## 39   17  Binomial_30_0.17 6.612776e-07
## 40   18  Binomial_30_0.17 9.551787e-08
## 41   19  Binomial_30_0.17 1.206542e-08
## 42   20  Binomial_30_0.17 1.327196e-09
## 43    0  Binomial_40_0.12 4.789852e-03
## 44    1  Binomial_40_0.12 2.737058e-02
## 45    2  Binomial_40_0.12 7.624663e-02
## 46    3  Binomial_40_0.12 1.379701e-01
```

```
## 47   4  Binomial_40_0.12 1.823176e-01
## 48   5  Binomial_40_0.12 1.875267e-01
## 49   6  Binomial_40_0.12 1.562722e-01
## 50   7  Binomial_40_0.12 1.084338e-01
## 51   8  Binomial_40_0.12 6.389849e-02
## 52   9  Binomial_40_0.12 3.245638e-02
## 53  10  Binomial_40_0.12 1.437354e-02
## 54  11  Binomial_40_0.12 5.600080e-03
## 55  12  Binomial_40_0.12 1.933361e-03
## 56  13  Binomial_40_0.12 5.948803e-04
## 57  14  Binomial_40_0.12 1.638956e-04
## 58  15  Binomial_40_0.12 4.058367e-05
## 59  16  Binomial_40_0.12 9.058855e-06
## 60  17  Binomial_40_0.12 1.826996e-06
## 61  18  Binomial_40_0.12 3.334992e-07
## 62  19  Binomial_40_0.12 5.516529e-08
## 63  20  Binomial_40_0.12 8.274793e-09
## 64   0 Binomial_100_0.05 5.920529e-03
## 65   1 Binomial_100_0.05 3.116068e-02
## 66   2 Binomial_100_0.05 8.118177e-02
## 67   3 Binomial_100_0.05 1.395757e-01
## 68   4 Binomial_100_0.05 1.781426e-01
## 69   5 Binomial_100_0.05 1.800178e-01
## 70   6 Binomial_100_0.05 1.500149e-01
## 71   7 Binomial_100_0.05 1.060255e-01
## 72   8 Binomial_100_0.05 6.487089e-02
## 73   9 Binomial_100_0.05 3.490130e-02
## 74  10 Binomial_100_0.05 1.671588e-02
## 75  11 Binomial_100_0.05 7.198228e-03
## 76  12 Binomial_100_0.05 2.809834e-03
## 77  13 Binomial_100_0.05 1.001075e-03
## 78  14 Binomial_100_0.05 3.274191e-04
## 79  15 Binomial_100_0.05 9.880016e-05
## 80  16 Binomial_100_0.05 2.762505e-05
## 81  17 Binomial_100_0.05 7.184222e-06
## 82  18 Binomial_100_0.05 1.743539e-06
## 83  19 Binomial_100_0.05 3.960394e-07
## 84  20 Binomial_100_0.05 8.441893e-08
## 85   0           Poisson 6.737947e-03
## 86   1           Poisson 3.368973e-02
## 87   2           Poisson 8.422434e-02
## 88   3           Poisson 1.403739e-01
## 89   4           Poisson 1.754674e-01
## 90   5           Poisson 1.754674e-01
## 91   6           Poisson 1.462228e-01
## 92   7           Poisson 1.044449e-01
## 93   8           Poisson 6.527804e-02
## 94   9           Poisson 3.626558e-02
## 95  10           Poisson 1.813279e-02
## 96  11           Poisson 8.242177e-03
## 97  12           Poisson 3.434240e-03
## 98  13           Poisson 1.320862e-03
## 99  14           Poisson 4.717363e-04
## 100 15           Poisson 1.572454e-04
```

```
## 101 16          Poisson 4.913920e-05
## 102 17          Poisson 1.445271e-05
## 103 18          Poisson 4.014640e-06
## 104 19          Poisson 1.056484e-06
## 105 20          Poisson 2.641211e-07
```

```
library(ggplot2)
library(RColorBrewer) # Color palettes

ggplot(df_plot, aes(x = X, y = value, color = variable)) +
  geom_line() +
  scale_color_manual(values = c(brewer.pal(4, "PRGn"), "red")) +
  labs(
    title = "Binomial vs Poisson PMF",
    x = "x",
    y = "P(X = x)"
  )
```



Exercise 3

A

Generate N=1000 random numbers from a binomial distribution with n=9 trials and p=0.8. Thus each of the 1000 random numbers will be an integer between 0 and 9.

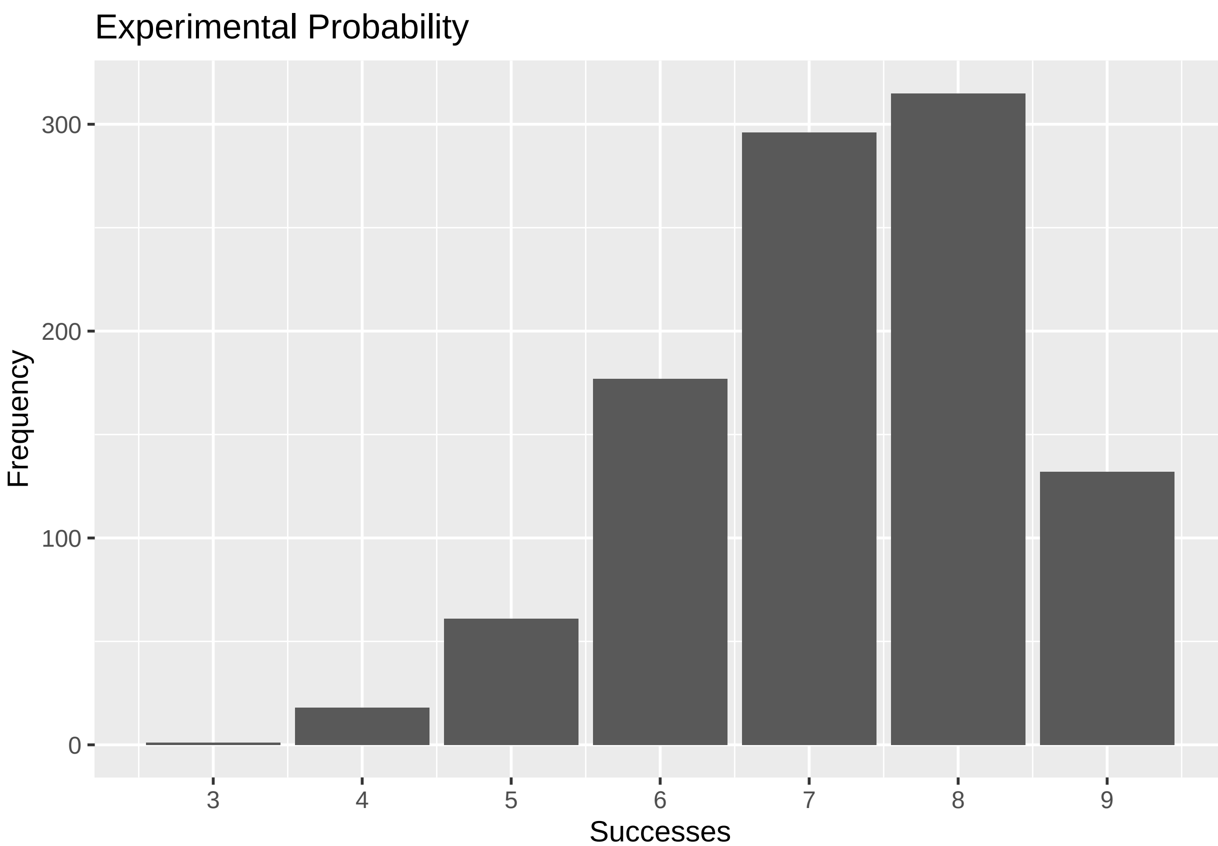
```
set.seed(123)
n <- 9
p <- 0.8
```

```r
t <- 1000

valori <- rbinom(t, n, p)
```

B

Plot the experimental probability using the geom_bar() function.

```r
df <- data.frame(x = valori)

ggplot(df, aes(x = valori)) +
  geom_bar() +
  scale_x_continuous(breaks = 0:9) +
  labs(
    title = "Experimental Probability",
    x = "Successes",
    y = "Frequency"
  )
```

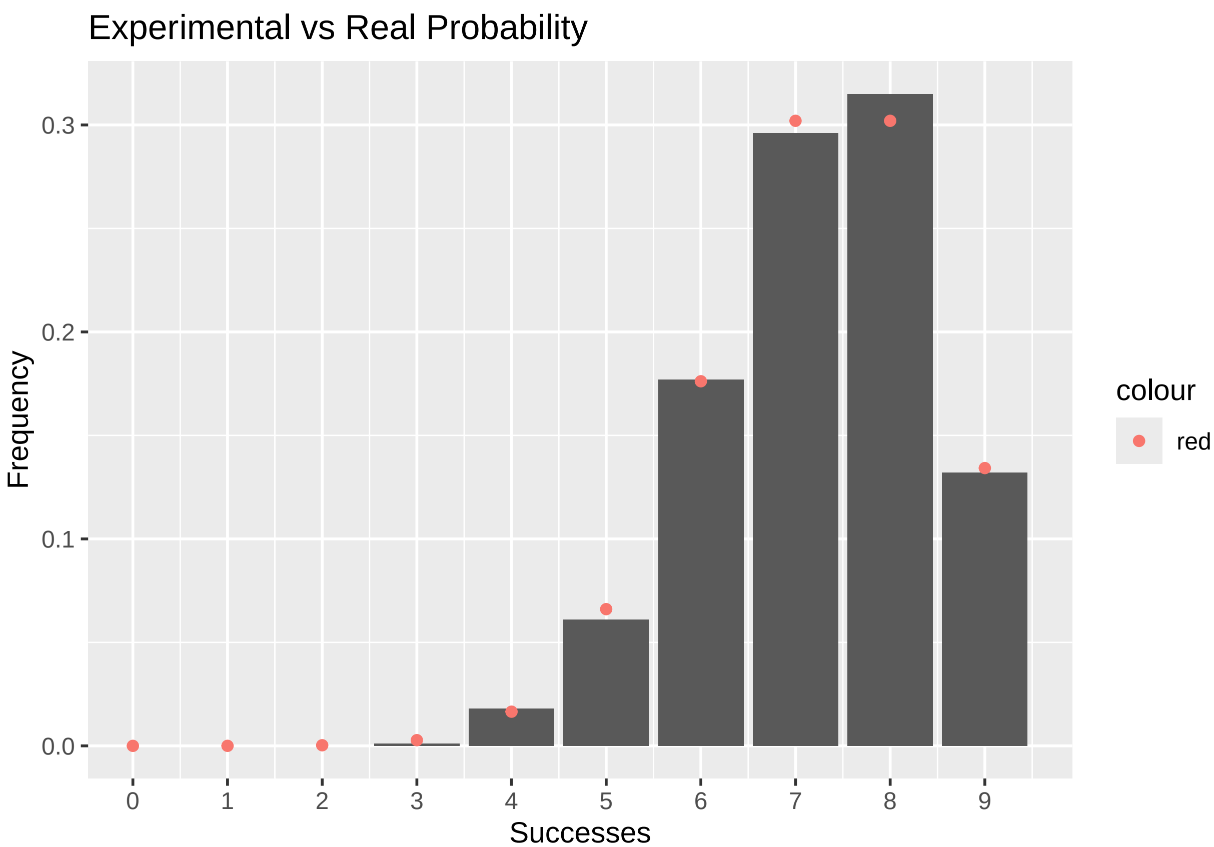


C

For each value of the x-axis obtained in the previous plot, compute the real probability mass function and add it in the plot as red dots using geom_point().

```r
p_teo <- data.frame(p = dbinom(0:9, n, p))

ggplot(df, aes(x = valori)) +
  geom_bar(aes(y = after_stat(prop))) +
```

```r
  geom_point(
    data = p_teo,
    aes(
      x = 0:9, y = p,
      color = "red"
    ),
  ) +
  scale_x_continuous(breaks = 0:9) +
  labs(
    title = "Experimental vs Real Probability",
    x = "Successes",
    y = "Frequency"
  )
```



Exercise 4

Suppose the number of customers visiting a retail store follows a Poisson distribution with a mean of 5 customers per hour. (hint: $X \sim Poisson(\lambda = 5)$)

Find the probability that in a randomely chosen hour, there will be:

A

No customers

```r
lambda <- 5
dpois(0, lambda)
```

```
## [1] 0.006737947
```

B

At least 3 customers

```r
ppois(2, lambda, lower.tail = FALSE)
```

```
## [1] 0.875348
```

C

Exactly 7 customers

```r
dpois(7, lambda)
```

```
## [1] 0.1044449
```

D

Assuming that each customer buys something that costs a price from 10€ to 50€. Which is the expected value of a customer expense? (hint: Y = customer expense ~ Uniform(10, 50))

```r
a <- 10
b <- 50

e_y <- (a + b) / 2
e_y
```

```
## [1] 30
```

E

How many customers are expected in 8 working hours?

```r
h <- 8

e_x <- lambda * h
e_x
```

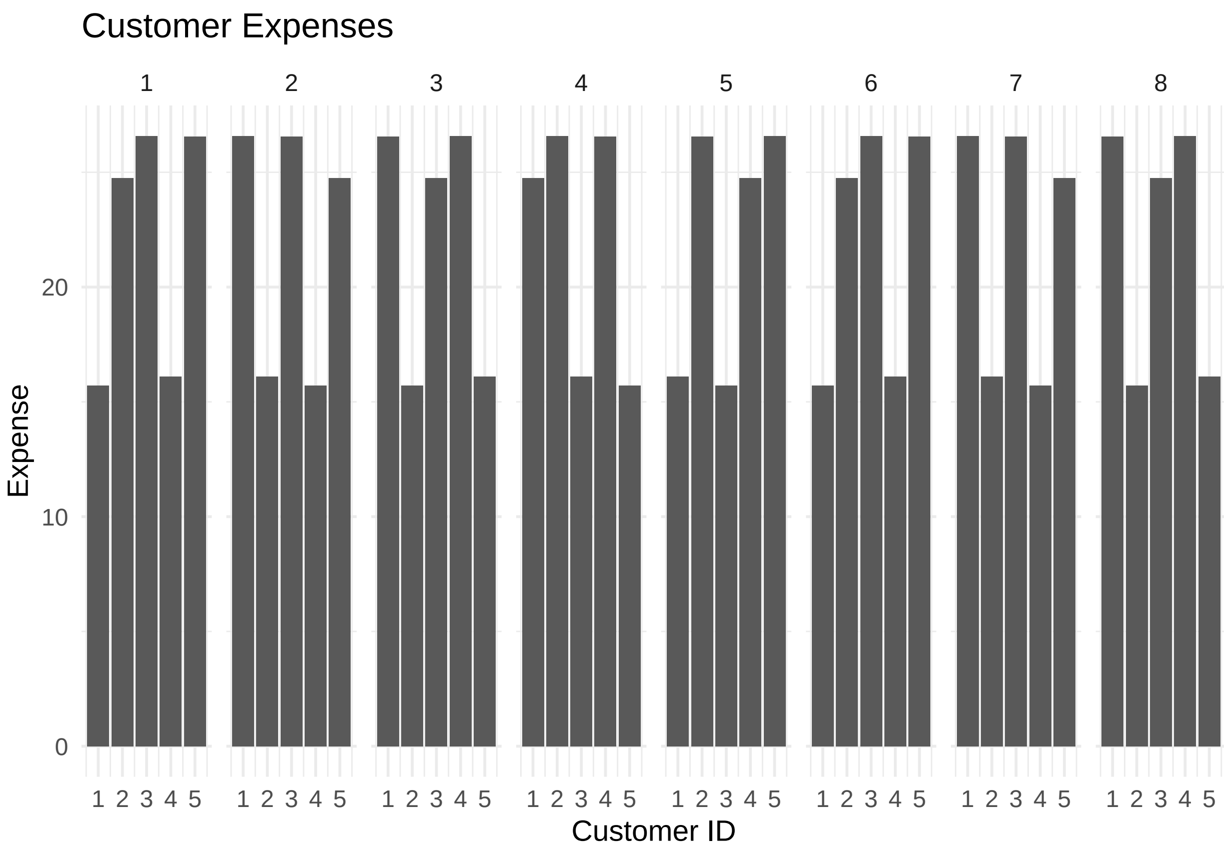
```
## [1] 40
```

F

Using rpois() and runif() functions, simulate the customers and their expenses in a normal day made of 8 working hours. Represent the data as similar as possible to the plot below (hints: prepare a data.frame containing as many rows as the number of customers of the entire day. For each customer, store its ID number, the hour, and the money he/she spent. Use facet_grid with scales = "free_x". Search on google how to do the rest).

```r
t <- 5
h <- 8

set.seed(123)
customers <- rpois(h * t, lambda)
expenses <- runif(t, 10, 50)

df <- data.frame(
```

```r
  customer = rep(1:t, each = h),
  hour = rep(1:h, t),
  expense = expenses
)

# Plot each hour in a separate facet
ggplot(df, aes(x = customer, y = expense)) +
  geom_col() +
  facet_grid(~hour, scales = "free_x") +
  labs(
    title = "Customer Expenses",
    x = "Customer ID",
    y = "Expense"
  ) +
  theme_minimal()
```



Exercise 5

In a survey of a sample of 500 university students, they were asked if they know how to program in R. The collected data showed that 240 students know how to program in R. Consider the discrete random variable X, which indicates the number of students in a random sample of 50 students who know how to program in R.

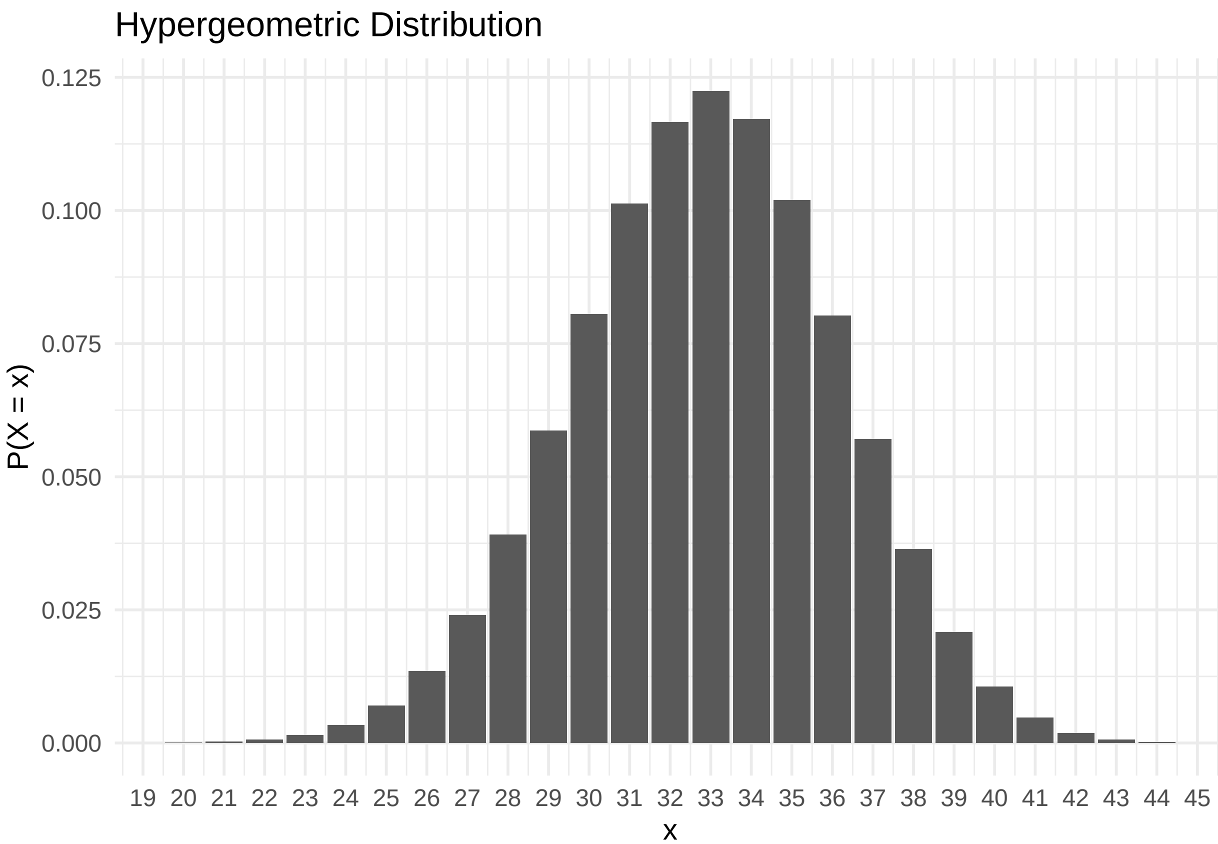
A

Compute the probability mass function of X and draw a plot.

```r
students <- 500
know_r <- 240
sample_size <- 50

dhyper(0:50, students, students - know_r, sample_size)
```

```
##  [1] 1.712717e-25 2.029286e-23 1.170237e-21 4.377677e-20 1.194605e-18
##  [6] 2.535451e-17 4.357806e-16 6.235766e-15 7.579816e-14 7.946693e-13
## [11] 7.271586e-12 5.862735e-11 4.197005e-10 2.684693e-09 1.542586e-08
## [16] 7.996764e-08 3.754012e-07 1.600830e-06 6.217258e-06 2.203976e-05
## [21] 7.144235e-05 2.120738e-04 5.771781e-04 1.441491e-03 3.305727e-03
## [26] 6.963691e-03 1.347683e-02 2.395881e-02 3.911283e-02 5.859870e-02
## [31] 8.049996e-02 1.012849e-01 1.165483e-01 1.224346e-01 1.171659e-01
## [36] 1.018762e-01 8.023788e-02 5.703308e-02 3.642637e-02 2.079578e-02
## [41] 1.054554e-02 4.713777e-03 1.839816e-03 6.196420e-04 1.773653e-04
## [46] 4.228946e-05 8.169881e-06 1.228289e-06 1.347904e-07 9.601336e-09
## [51] 3.330925e-10
```

```r
df <- data.frame(x = 20:44, y = dhyper(
  20:44,
  students,
  students - know_r,
  sample_size
))

ggplot(df, aes(x = x, y = y)) +
  geom_col() +
  scale_x_continuous(breaks = 0:50) +
  labs(
    title = "Hypergeometric Distribution",
    x = "x",
    y = "P(X = x)"
  ) +
  theme_minimal()
```



B

Calculate the probability that at least 30 students in the sample of 50 know how to program in R.

```
1 - phyper(29, students, students - know_r, sample_size)
```

```
## [1] 0.8522509
```

C

Calculate the expected value of X.

```
students * sample_size / students
```

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