Assignment_2

Abbas Zal

2023-04-24

```r
#adding library
library(ggplot2)
library(gridExtra)
```

Exercise 1 - Discrete random variable

the probability distribution function of a discrete variable k is given by the following:

```r
pk <- function(k) {
  if (k >= 1 & k <= 5) {
    return(k/15)
  } else {
    return(0)
  }
}
```

1_1

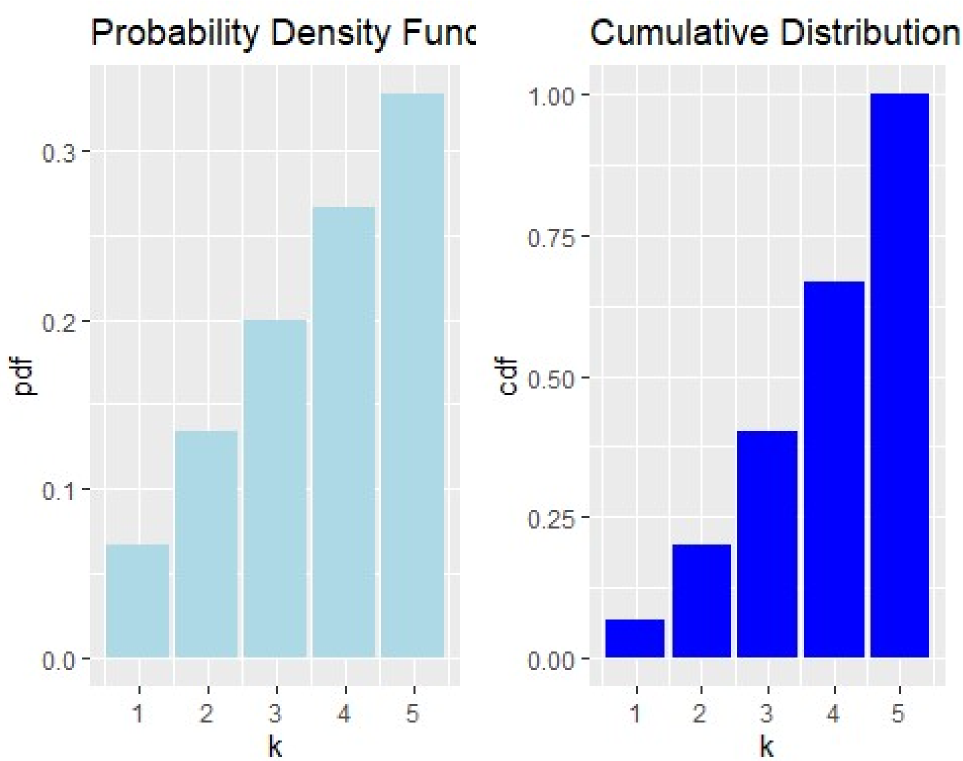
Now we want to write the R probability functions for the probability density and cumulative distribution functions:

```r
# Define probability density function
dprob <- Vectorize(pk)

# Define cumulative distribution function
pprob <- function(k) {
  if(k >= 1 & k < 2) {
    return(pk(1))
  } else if (k >= 2 & k < 3) {
    return(pk(1) + pk(2))
  } else if (k >= 3 & k < 4) {
    return(pk(1) + pk(2) + pk(3))
  } else if (k >= 4 & k < 5) {
    return(pk(1) + pk(2) + pk(3) + pk(4))
  } else if (k == 5) {
    return(1)
  } else{
    return(0)
  }
}
```

1_2

Writing code for plotting and showing the pdf and cdf:

```
k <- 1:5

pdf <- dprob(k)
cdf <- cumsum(pdf)

pdf_df <- data.frame(k = k, pdf = pdf)
cdf_df <- data.frame(k = k, cdf = cdf)

# Create PDF plot
pdf_plot <- ggplot(pdf_df, aes(x = k, y = pdf)) +
  geom_bar(stat = "identity", fill = "lightblue") +
  labs(title = "Probability Density Function (PDF)", x = "k")

# Create CDF plot
cdf_plot <- ggplot(cdf_df, aes(x = k, y = cdf)) +
  geom_bar(stat = "identity", fill = "blue") +
  labs(title = "Cumulative Distribution Function (CDF)", x = "k")

# Display both plots side by side
grid.arrange(pdf_plot, cdf_plot, ncol = 2)
```



1_3

Computing the mean value and variance of the probability distribution:

```
mean_pk = sum(pdf * k)
variance_pk = sum(pdf * k^2 - (pdf * k)^2)
mean_pk

## [1] 3.666667

variance_pk

## [1] 10.64889
```

1_4

The expected value E[k (6 – k)]

```
expected_value <- 0
for(k in 1:5){
  r <- k * (6 - k) * pk(k)
  expected_value <- r + expected_value
}
expected_value

## [1] 7
```

1_5

Then we would like writing the R function that allows to sample random numbers from the probability distribution:

```
rand_function <- function(n){
  k <- 1:5
  samples <- sample(k, size = n, replace = TRUE, prob = sapply(1:5, pk))
  return(samples)
}
```

For example :

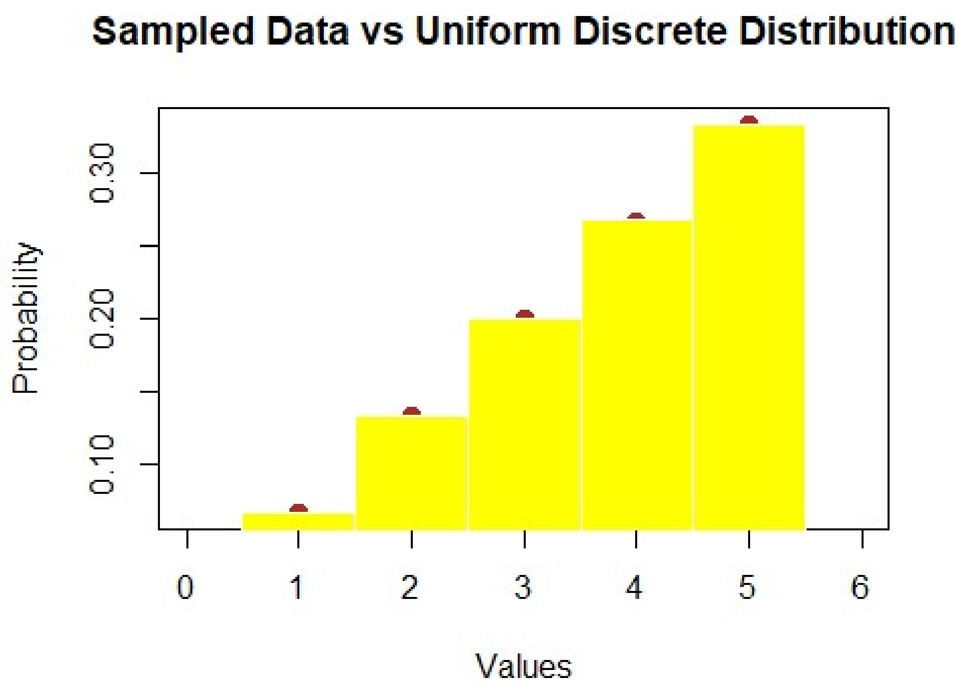
```
#ten rand number
rand_function(10)

##  [1] 5 5 1 5 3 5 4 5 2 4
```

1_6

Using the implemented function (point (5)), sample 100000 random numbers from this distribution and plot them in a graph showing the distribution of the numbers superimposed to the pdf (normalize properly the plots with random numbers):

```
set.seed(123)
n <- 100000
y<- rand_function(n)
x <- numeric(n)
for (i in 1:n) {
  x[i] <- pk(y[i])
```

```
}

plot(y, dprob(y), type = "h", col = "brown", lwd = 10, xlim = c(0,6),
     ylab = "Probability", xlab = "Values", main = "Sampled Data vs Uniform
Discrete Distribution")
hist(y, breaks = seq(0 - 0.5 , 6+ 0.5, by = 1), col = "yellow", border =
"white" , freq = FALSE ,add = TRUE)
```



Exercise 2 - Continuous random variable

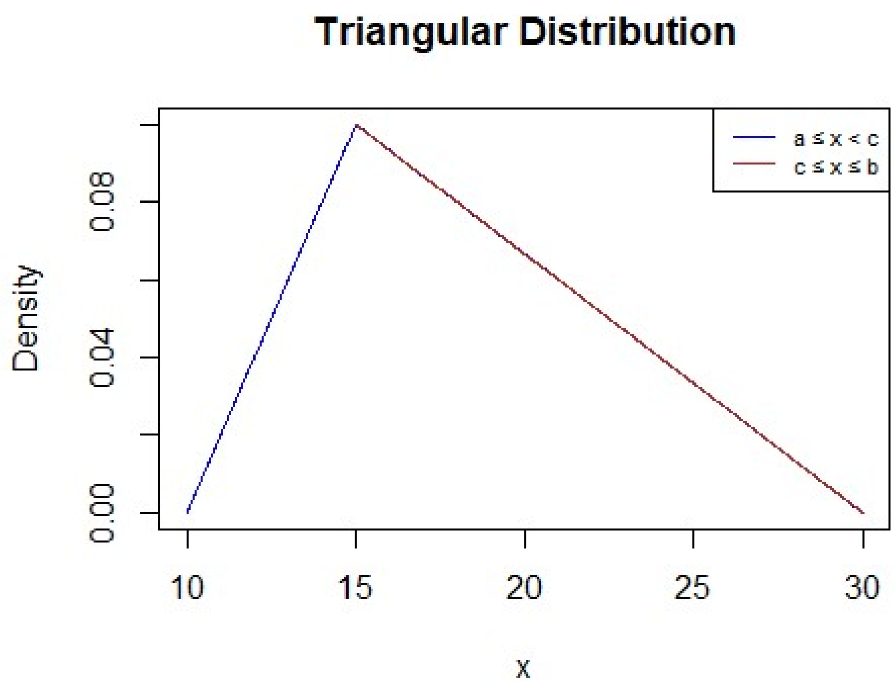
The triangular distribution, in the interval (a, b), is given by the following:

```
# Define triangular distribution function
triangular <- function(x, a, b, c) {
  ifelse(x < a | x > b, 0,
         ifelse(x < c, 2*(x-a)/((b-a)*(c-a)),
                2*(b-x)/((b-a)*(b-c))))
}
```

Now, we want to plot the function, given the interval (a, b):

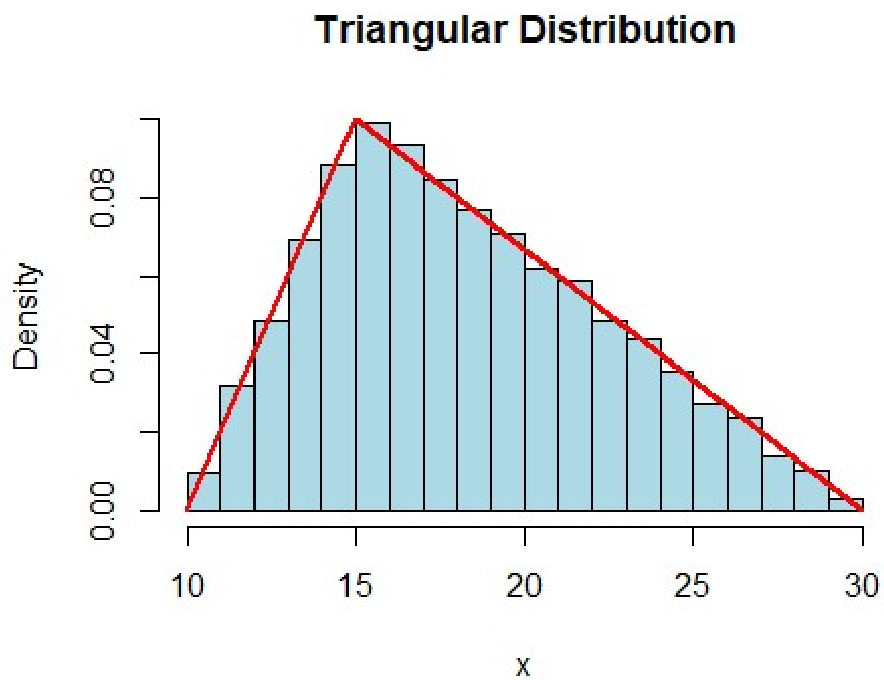
```
a <- 10
b <- 30
c <- 15

# Plot density curve
curve(triangular(x, a, b, c), from = a, to = b, n = 1000, col = "blue",
```

```
     xlab = "x", ylab = "Density", main = "Triangular Distribution")
curve(triangular(x, a, b, c), col = "brown", add = TRUE, from = c, to = b, n
= 1000)
#segments(x_coord, 0, x_coord, y_coord, col = "green", lty = "dashed")
legend("topright", legend=c("a ≤ x < c", "c ≤ x ≤ b"),
       col=c("blue","brown"), lty=c(1,1),cex = 0.7)
```



Writing an algorithm to generate random numbers from the triangular distribution:

```
random_triangular <- function(n, a, b, c) {
  U <- runif(n)
  x <- ifelse(U < (c - a) / (b - a),
              a + sqrt(U * (b - a) * (c - a)),
              b - sqrt((1 - U) * (b - a) * (b - c)))
  return(x)
}

set.seed(123)
n <- 10000
x <- random_triangular(n,a,b,c)

hist(x, prob = TRUE,col = "lightblue", main = "Triangular Distribution",
     xlab = "x", ylab = "Density")
curve(triangular(x, a, b, c), add = TRUE, col = "red",from = a, to = b, n =
10000, lwd = 2)
```



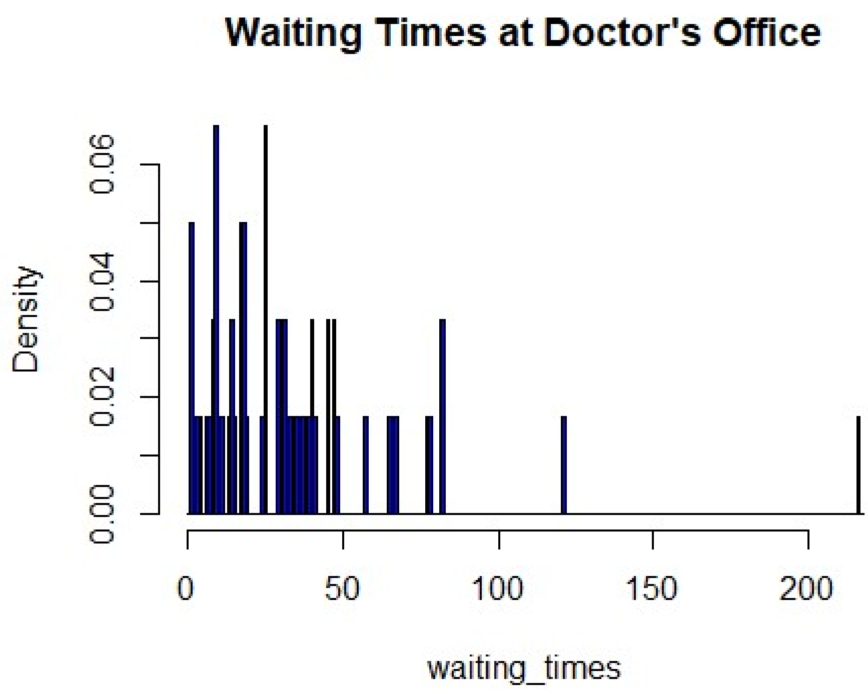
Exercise 3

The waiting time, in minutes, at the doctor's is about 30 minutes, and the distribution follows an exponential pdf with rate 1/30. Now we want to simulate the waiting time for 60 people at the doctor's office and plot the relative histogram:

```
set.seed(123)

# Generate 60 waiting times from the exponential distribution
waiting_times <- rexp(60, rate = 1/30)
waiting_times

##  [1]  25.3037178  17.2983081  39.8716460   0.9473208   1.6863293
9.4950365
##  [7]   9.4268188   4.3580041  81.7870939   0.8746034  30.1449017
14.4064418
## [13]   8.4304088  11.3135349   5.6485212  25.4935839  46.8961062
14.3628125
## [19]  17.7280451 121.2303513  25.2944919  28.9761363  44.5582738
40.4413346
## [25]  35.0558695  48.1755703  44.9022861  47.1195764   0.9530323
17.9354907
## [31]  65.0351924  15.1984719   7.7867345  77.9067635  36.8707720
23.7204528
## [37]  18.8784023  37.6392301  17.6605393  33.8787010  12.6109441
216.3302273
## [43]  25.3716590   6.7662602  33.0101645  67.4491708  40.9120290
```

```
17.2917500
## [49]  81.7582755  39.3648913   2.7177405   9.1861155  32.0163921
9.4054877
## [55]  29.2392045  56.6346995  16.9376581  77.3088399  31.4308724
30.7332403
```

```
breaks = seq(-0.5 , max(waiting_times) + 1 + 0.5, by = 1.0)
# Plot the relative histogram of the waiting times
hist(waiting_times,breaks = breaks, freq = FALSE, main = "Waiting Times at
Doctor's Office", col = "blue")
```



The probability for waiting time less than 12 minutes:

```
t <- 12
waiting_times[which(waiting_times <= 12)]

##  [1]  0.9473208  1.6863293  9.4950365  9.4268188  4.3580041  0.8746034
##  [7]  8.4304088 11.3135349  5.6485212  0.9530323  7.7867345  6.7662602
## [13]  2.7177405  9.1861155  9.4054877

p <-length(which(waiting_times <= 12)) / length(waiting_times)

# Print the probability
cat("The probability that a person will wait for less than", t,
    "minutes is", round(p, 4)*100, "%.")

## The probability that a person will wait for less than 12 minutes is 25 %.
```

Evaluate the average waiting time from the simulated data and compare it with the expected value:

```
# Calculate the average waiting time
avg_waiting_time <- mean(waiting_times)

# Print the results
cat("The average waiting time from the simulated data is",
round(avg_waiting_time, 2), "minutes.")

## The average waiting time from the simulated data is 33.19 minutes.

# Set the parameters
rate <- 1/30 # rate parameter for exponential distribution

# Calculate the expected value
expected_value <- 1/rate

# Print the results
cat("The expected value of the waiting time is", expected_value, "minutes.")

## The expected value of the waiting time is 30 minutes.
```

What is the probability for waiting more than one hour before being received ?

```
t <- 60
waiting_times[which(waiting_times <= 60)]

##  [1] 25.3037178 17.2983081 39.8716460  0.9473208  1.6863293  9.4950365
##  [7]  9.4268188  4.3580041  0.8746034 30.1449017 14.4064418  8.4304088
## [13] 11.3135349  5.6485212 25.4935839 46.8961062 14.3628125 17.7280451
## [19] 25.2944919 28.9761363 44.5582738 40.4413346 35.0558695 48.1755703
## [25] 44.9022861 47.1195764  0.9530323 17.9354907 15.1984719  7.7867345
## [31] 36.8707720 23.7204528 18.8784023 37.6392301 17.6605393 33.8787010
## [37] 12.6109441 25.3716590  6.7662602 33.0101645 40.9120290 17.2917500
## [43] 39.3648913  2.7177405  9.1861155 32.0163921  9.4054877 29.2392045
## [49] 56.6346995 16.9376581 31.4308724 30.7332403

p <- 1 - length(which(waiting_times <= 60)) / length(waiting_times)

# Print the probability
cat("The probability of waiting more than", t, "minutes is", round(p, 4)*100,
"%.")

## The probability of waiting more than 60 minutes is 13.33 %.
```

Exercise 4 - Multiple choices exams

The final exam of a course is given to the students in the format of a multiple choice written test: for each questions there are five possible alternatives, A student either knows the answer, or selects randomly the answer among the five possible choices. Now we are assuming p = 0.7 the probability that the student knows the answer, once a correct answer is given, what it the probability that the student really knew the correct answer ?

```
p_knows <- 0.7
p_doesnt_know <- 1 - p_knows
p_correct_given_knows <- 1
p_correct_given_doesnt_know <- 1/5
p_correct <- p_correct_given_knows * p_knows + p_correct_given_doesnt_know *
p_doesnt_know
p_knows_given_correct <- p_correct_given_knows * p_knows / p_correct
p <- p_knows_given_correct
# Print the probability
cat("The probability that the student really knew the correct answer is",
round(p, 4)*100, "%.")

## The probability that the student really knew the correct answer is 92.11
%.
```

Exercise 5 - Waiting time

Starting from 5:00 in the morning, every half an hour there is a train from Milano Centrale to Roma Termini. We assume there is always an available seat on a train leaving from Milano .Assuming a person arrives at a random time between 10:45 and 11:45 and compute the probability that she has to wait:

a) at most 10 minutes

b) at least 15 minutes

c) what is the average time spent waiting ?

For section (a) if we want to consider waiting time at most 10 minutes, person must arrive at 10:50 for 11:00 train or 11:20 for 11:30 train. It means the overall time is equal with 20 minutes from 10:45 to 11:45.

```
p <- 20 /60
# Print the result
cat("The probability that she has to wait at most 10 minutes is", round(p,
4)*100 , "%")

## The probability that she has to wait at most 10 minutes is 33.33 %
```

For section (b) if we want to consider waiting time at least 15 minutes, person must arrive from 11:00 to 11:15 for 11:30 train or between 11:30 and 11:45 for 12:00 train. It means the overall time is equal with 30 minutes from 10:45 to 11:45.

```
p <- 30 /60
# Print the result
cat("The probability that she has to wait at least 15 minutes is", round(p,
4)*100 , "%")
```

```
## The probability that she has to wait at least 15 minutes is 50 %
```

Exercise 6 - stock investment

The annual return rate for a specific stock on the market is a normal variable with a 10% mean and a 12% standard deviation. Mr X decides to buy 200 share of that specific stock at a price of 85EUR per share. what is the probability that after a year his net profit from the investment is at least 800EUR ?

```
# Define the parameters
mean <- 0.1  # Mean annual return rate
sd <- 0.12   # Standard deviation of annual return rate
n_shares <- 200  # Number of shares
price_per_share <- 85  # Price per share



# Calculate the probability that the net profit is at least 800 euros
prob <- 1 - pnorm(800/ (n_shares*price_per_share)  , mean , sd )


# Print the result
cat("The probability that Mr. X's net profit is at least 800 EUR is",
round(prob, 4)*100 , "%")

## The probability that Mr. X's net profit is at least 800 EUR is 67.05 %
```