

# Assignment 2

## Exercise 1: Discrete random variable

- Write the R functions for the probability density and cumulative distribution functions, using the R naming convention.
- Produce two plots showing the pdf and cdf, separately.
- Compute the mean value and variance of the probability distribution using R. 4. Generate a sample of random numbers from this distribution and show them in an histogram. Evaluate the sample mean.

```
# Parameters for the poisson distribution
lambda_vec <- c(1.4,4,6, 8)
k <- 1:13

integrate_step <- function(r, lower, upper){
  integral = 0
  for(x in lower:upper-1){
    integral = integral+f(x)
  }
  return(integral)
}

dzeropois <- function(k, lambda){
  #to handle float numbers
  k = trunc(k)
  return(lambda^k*exp(- lambda)/factorial(k)*(1-exp(- lambda)))
}

#designed to handle vectors of q in input
pzeropois <- function(q, lambda) {
  #to handle float numbers
  q = as.integer(trunc(q))
  P_cum <- numeric(length(q))
  for (i in 1:length(q)) {
    k <- 1:q[i]
    P_cum[i] <- sum(dzeropois(k, lambda))
  }
  return(P_cum)
}

# Function used to invert the dzeropois
threshold_search <- function(r, guess, threshold, lambda) {
  x <- as.integer(trunc(guess))
  while(TRUE) {
    if(f(x, lambda) - threshold <= 0) {
      if(f(x + 1, lambda) - threshold >= 0)
        return(x+1)
      else
        x <- x + 1
    }
    else if(x<1)
      x <- x - 1
    else
      return(x)
  }
}

# Function to generate random samples from Poisson distribution
rzeropois <- function(n, lambda) {
  p <- runif(n, min = 1e-20, max = 1-1e-10)
  x <- sapply(p, function(probability) {
    threshold_search(pzeropois, lambda=1, probability, lambda)
  })
  return(x)
}

poisson_pdf <- sapply(lambda_vec, function(L) {dzeropois(x, L)})
poisson_cum <- sapply(lambda_vec, function(L) {dzeropois(x, L)})

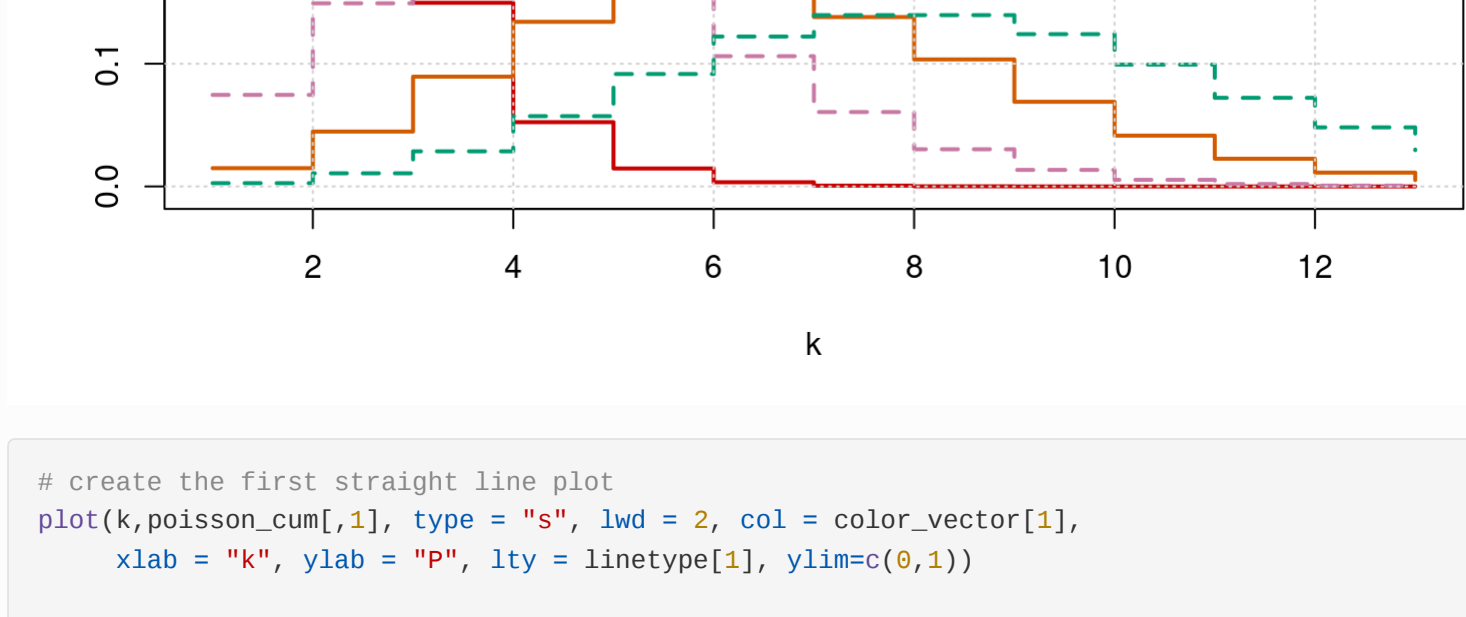
linetype <- 2*(1:length(lambda_vec))%2

# create the first straight line plot
plot(k,poisson_pdf[,1], type = "s", lwd = 2, col = color_vector[1],
      xlab = "k", ylab = "PDF", lty = linetype[1])

# create all the others
for (i in 2:length(lambda_vec)) {
  lines(k, poisson_pdf[, i], col= color_vector[i], type='s', lwd=2, lty = linetype[i])
}

grid()

# Create legend labels
legend_labels <- paste("lambda = ", lambda_vec)
# Add a legend
legend("topright", legend = legend_labels, col = color_vector[1:length(lambda_vec)], lty = linetype, 1)
# Add a title
title("Theoretical zero-truncated Poisson Distribution")
```

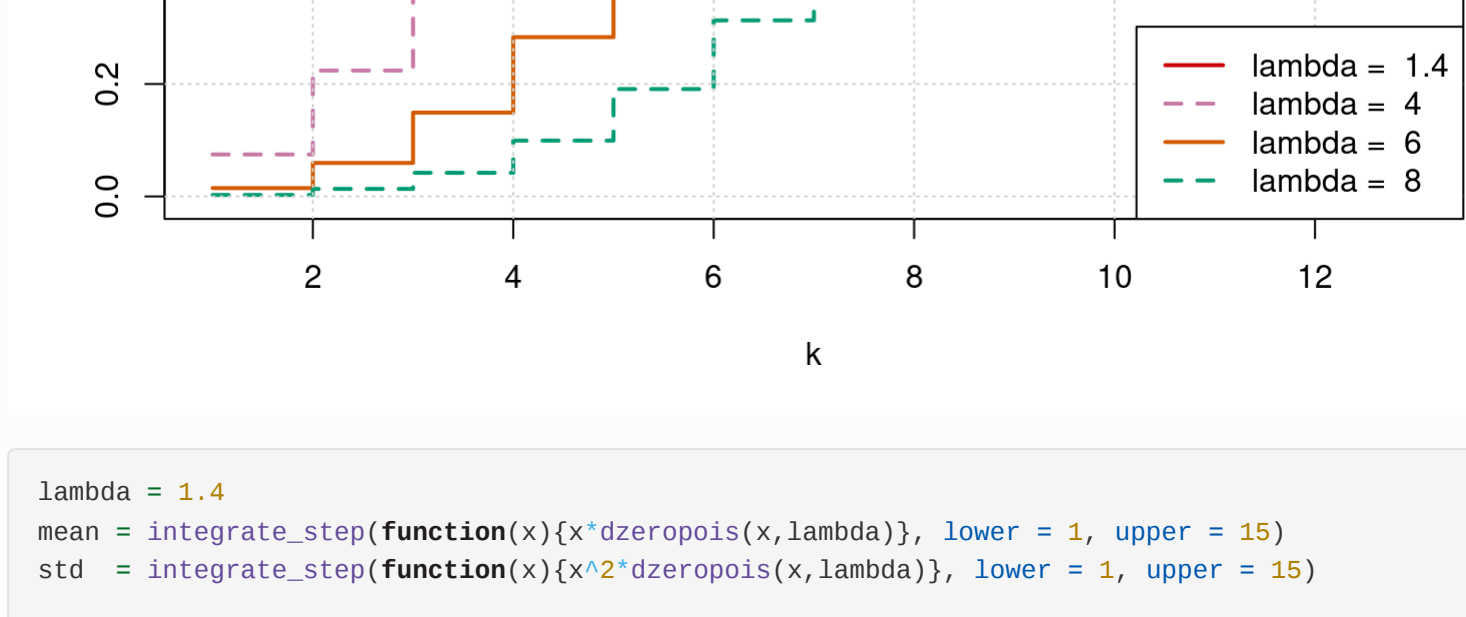


```
# create the first straight line plot
plot(k,poisson_cum[,1], type = "s", lwd = 2, col = color_vector[1],
      xlab = "k", ylab = "PF", lty = linetype[1], ylim=c(0,1))

# create all the others
for (i in 2:length(lambda_vec)) {
  lines(k, poisson_cum[, i], col= color_vector[i], type='s', lwd=2, lty = linetype[i], ylim=c(0,1))
}

grid()

# Create legend labels
legend_labels <- paste("lambda = ", lambda_vec)
legend("bottomright", legend = legend_labels, col = color_vector[1:length(lambda_vec)], lty = linetype)
title("Theoretical zero-truncated Poisson Cumulative Distribution")
```



```
lambda = 1.4
mean = integrate_step(function(x){x*dzeropois(x,lambda)}, lower = 1, upper = 15)
std = integrate_step(function(x){x^2*dzeropois(x,lambda)}, lower = 1, upper = 15)

writeLines(sprintf(
  "
  Taking lambda = 1.4 and truncating the sum at k=15 we can obtain:
  - The mean value is %.2f
  - The standard deviation is %.2f
",
  mean[1], std[1])
)
```

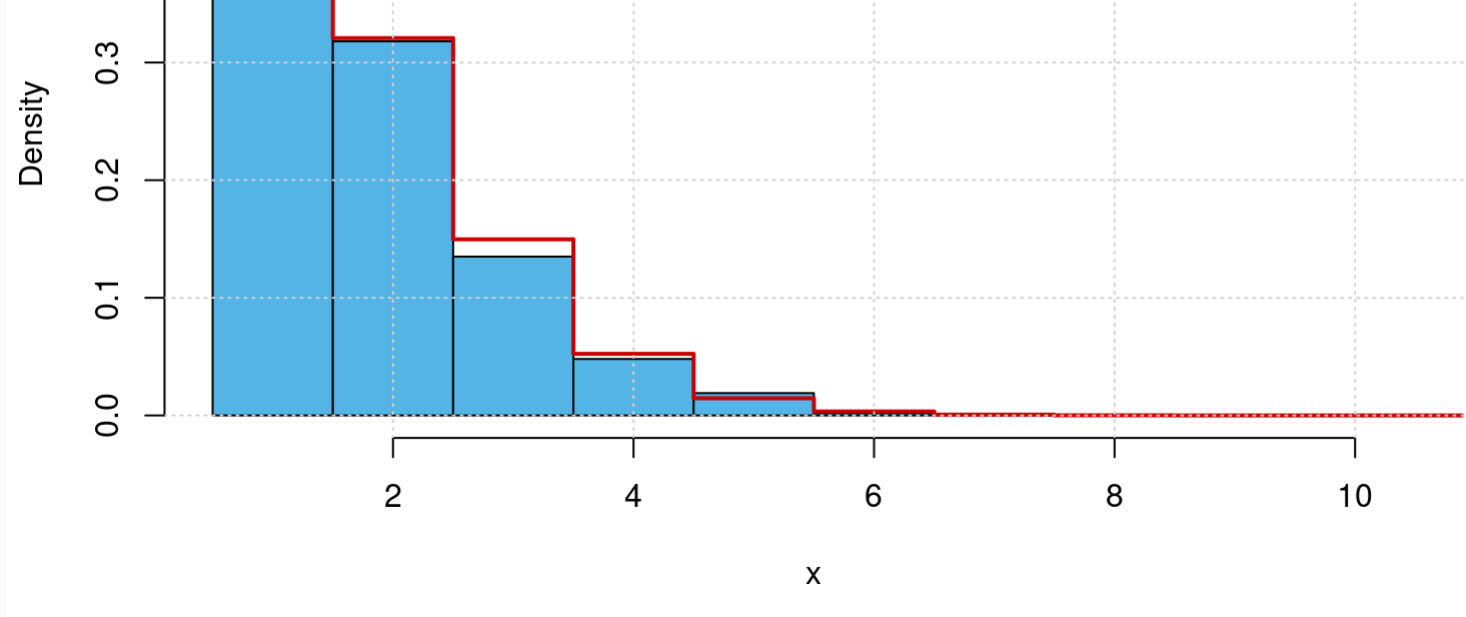
```
##
## Taking lambda = 1.4 and truncating the sum at k=15 we can obtain:
## - The mean value is 1.86
## - The standard deviation is 4.46
```

```
n = 1000
random_numbers <- rzeropois(n, lambda)

breaks <- seq(1 - 0.5, 10 + 0.5, by = 1)
#plot
hist(random_numbers, breaks=breaks, col = color_vector[5], xlab = "x", ylab = "Density", freq = FALSE,
      lines(k=0.5, poisson_pdf[,1], type='s', lwd = 2, col = color_vector[1], lty = linetype[1])

legend_labels <- paste("True distribution")
legend("topright", legend = legend_labels, col = color_vector[1], lty = linetype[1], lwd = 2)

grid()
title("Distribution of generated random numbers")
```



## Exercise 2: Continuous random variable

- Considering the energy distribution of CR muons at sea level:
  - Compute the normalization factor;
  - Plot the probability density function in R.
  - Plot the cumulative density function in R.
  - Compute the mean value using R
  - (Optional) Generate  $10^6$  random numbers from this distribution, show them in an histogram and superimpose the pdf (with a line or with a sufficient number of points).

```
dmuenergy <- function(E, N, E0, g){ ifelse(test = E<E0, yes = return(N), no = return(N*(E-E0)^1)/(1-g)
muenergy <- function(E, N, E0, g){ return(integrate(function (x) {sapply(x, function(E) {dmuenergy(
qmuenergy <- function(p, N, E0, g){ ifelse(p<N*E0, p/N, ((1-g)^(p/N*E0)-1)/(1-g)*E0-1) }

rmuenergy <- function(N, N, E0, g){
  uniform_sampling <- runif(n, min = 1.e-10, max = 1-1.e-10)
  return(sapply(uniform_sampling, function(p) qmuenergy(p, N, E0, g)))
}

normalize <- function(E0, g){ pmuenergy(Inf, 1, E0, g)^(-1) }
```

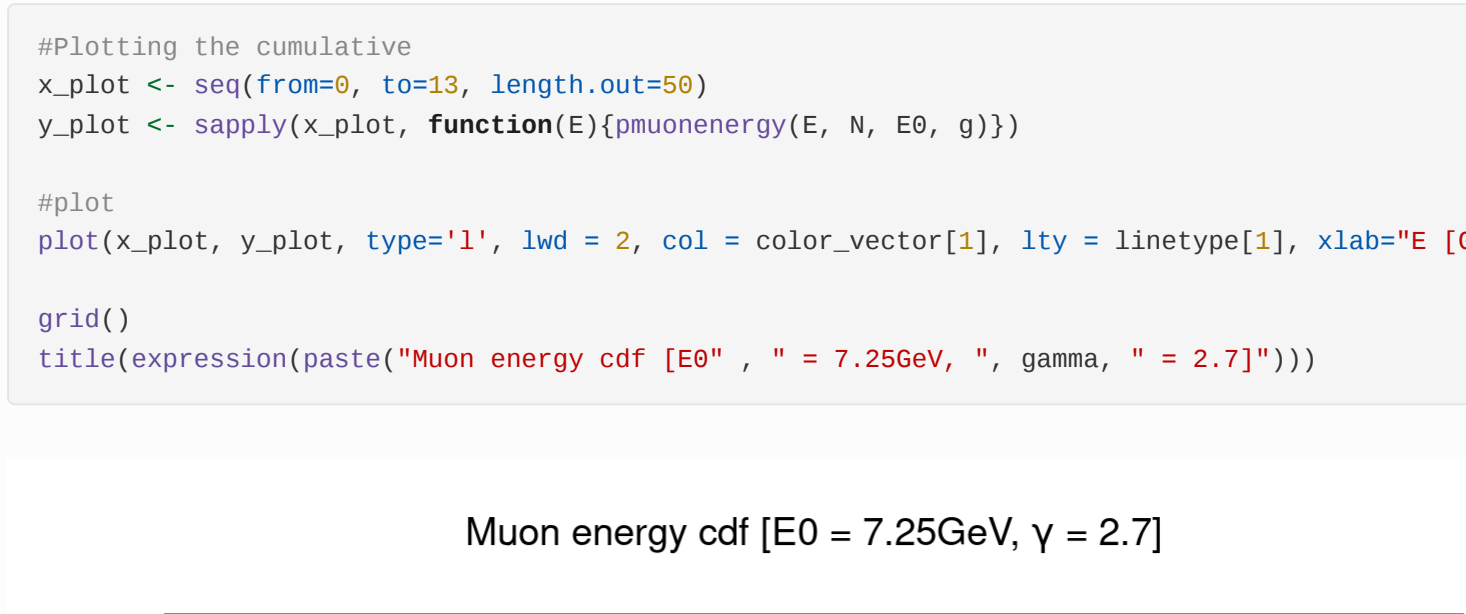
```
E0 <- 7.25 #GeV
g <- 2.7 #gamma factor

#Compute the normalization factor
N = normalize(E0, g)
print(sprintf("The normalization factor is %.2f", N))
```

```
## [1] "The normalization factor is 0.13"
```

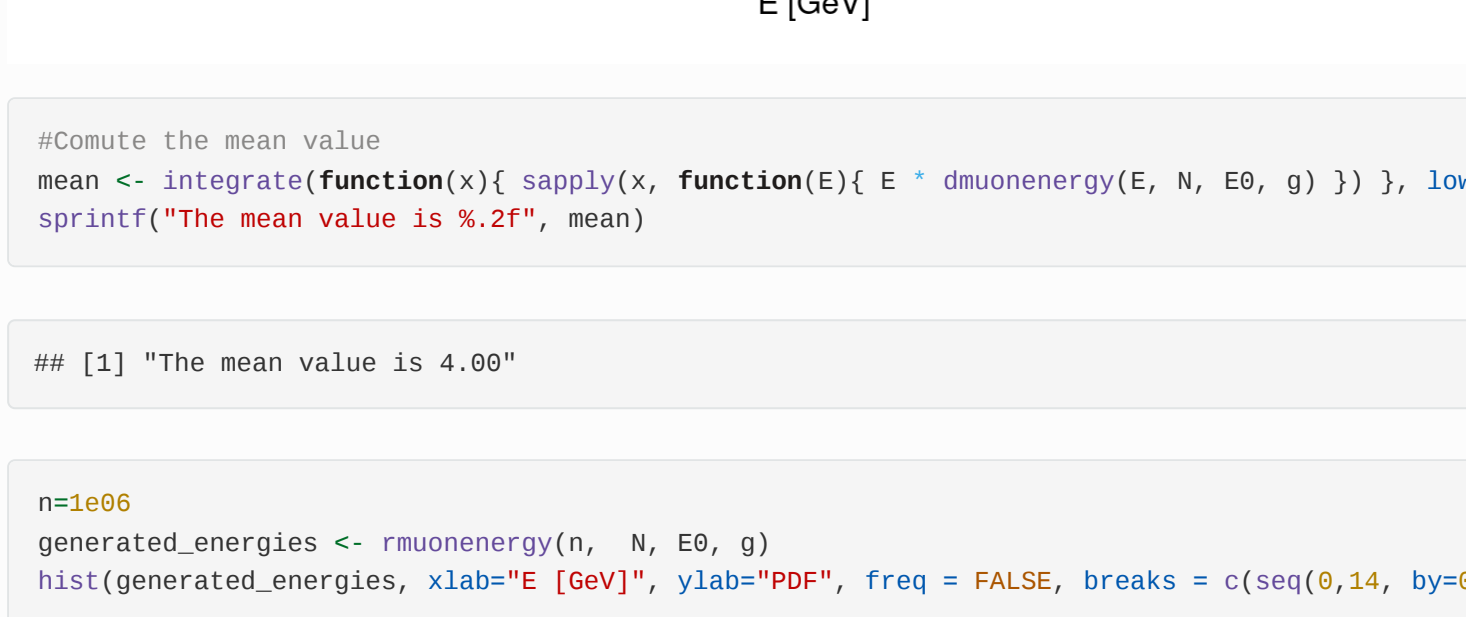
```
#Plotting the pdf
x_plot <- seq(from=0, to=13, length.out=50)
y_plot <- sapply(x_plot, function(E){dmuenergy(E, N, E0, g)})

plot(x_plot, y_plot, type='l', lwd = 2, col = color_vector[1], lty = linetype[1], xlab="E [GeV]", ylab=
grid()
title(expression(paste("Muon energy pdf [E0 = 7.25GeV, ", gamma, " = 2.7]")))
```



```
#Plotting the cumulative
x_plot <- seq(from=0, to=13, length.out=50)
y_plot <- sapply(x_plot, function(E){pmuenergy(E, N, E0, g)})

plot(x_plot, y_plot, type='l', lwd = 2, col = color_vector[1], lty = linetype[1], xlab="E [GeV]", ylab=
grid()
title(expression(paste("Muon energy cdf [E0 = 7.25GeV, ", gamma, " = 2.7]")))
```



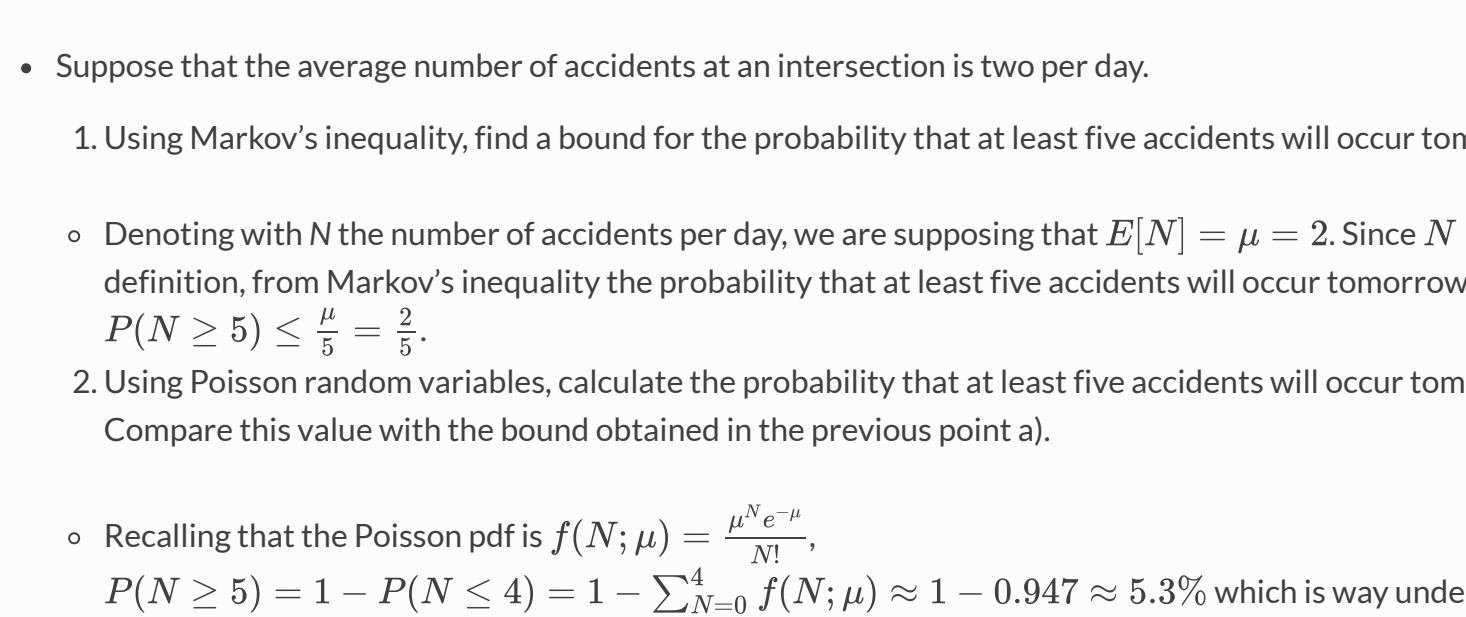
```
#Compute the mean value
mean <- integrate(function(x) {sapply(x, function(E) {E * dmuenergy(E, N, E0, g) } ) }, lower = 0, up
sprintf("The mean value is %.2f", mean)
```

```
## [1] "The mean value is 4.00"
```

```
n=1e6
generated_energies <- rmuenergy(n, N, E0, g)
hist(generated_energies, xlab="E [GeV]", ylab="PDF", freq = FALSE, breaks = c(seq(0,14, by=0.2), 15, n
#Plotting the pdf
x_plot <- seq(from=0, to=13, length.out=50)
y_plot <- sapply(x_plot, function(E){dmuenergy(E, N, E0, g)})

lines(x_plot, y_plot, type='l', lwd = 2, col = color_vector[1], lty = linetype[1])
legend_labels <- paste("Expected distribution")
legend("topright", legend = legend_labels, col = color_vector[1], lty = linetype[1], lwd = 2)

grid()
title(expression(paste("Generated muon energies pdf [E0 = 7.25GeV, ", gamma, " = 2.7]")))
```



## Exercise 3

- Suppose that the average number of accidents at an intersection is two per day.
  - Using Markov's inequality, find a bound for the probability that at least five accidents will occur tomorrow.
    - Denoting with  $N$  the number of accidents per day, we are supposing that  $E[N] = \mu = 2$ . Since  $N \geq 0$  by definition, from Markov's inequality the probability that at least five accidents will occur tomorrow is  $P(N \geq 5) \leq \frac{\mu}{5} = \frac{2}{5}$ .
  - Using Poisson random variables, calculate the probability that at least five accidents will occur tomorrow. Compare this value with the bound obtained in the previous point a).
    - Recalling that the Poisson pdf is  $f(N; \mu) = \frac{e^{-\mu} \mu^N}{N!}$ ,  $P(N \geq 5) = 1 - P(N \leq 4) = 1 - \sum_{k=0}^4 \frac{e^{-\mu} \mu^k}{k!}$ ,  $P(N \geq 5) \approx 1 - 0.947 \approx 5.3\%$  which is way under the upper bound of 40%.
  - Let the variance of the number of accidents be two per day. Using Chebyshev's inequality, find a bound on the probability that tomorrow at least five accidents will occur.
    - Considering  $\sigma^2 = 2$ , using Chebyshev's inequality  $P(N - 2 \geq 3) = P(N - \mu \geq 3) \leq \frac{\sigma^2}{9} = \frac{2}{9}$

## Exercise 4

The waiting period from the time a book is ordered until it is received is a random variable with mean seven days and standard deviation two days. If Helen wants to be 95% sure that she receives a book by certain date, how early should she order the book?

- Defining as  $T$  the random variable associated with the waiting period, let's  $E[T] = \mu = 7, \sigma^2 = 4$  the mean and the variance, the upper bounds defined by Markov and Chebyshev's inequalities are respectively:
  - $P(T < k) = 1 - P(T \geq k) = 1 - \frac{\mu}{k} \geq \frac{k-\mu}{k}$
  - $P(T < k) = 1 - P(T \geq k) = 1 - P(T - \mu \geq k - \mu) \geq 1 - P(T - \mu \geq k - \mu) \geq 1 - \frac{\sigma^2}{(k-\mu)^2}$

Plugging in the numbers we obtain the following inequalities:

$$P(T < k) \geq 0.95 \rightarrow \frac{k-7}{k} \geq 0.95. P(T < k) \geq 0.95 \rightarrow 1 - \frac{4}{(k-7)^2} \geq 0.95 \rightarrow (k-7)^2 \geq 80.$$

In particular, from the second inequality, we can extrapolate that to reach the 95% of confidence level, the lower bound to the waiting period is about 16 days.

## Exercise 5

An ordinary deck of 52 cards is divided randomly into 26 pairs. Using Chebyshev's inequality, find an upper bound for the probability that, at most, 10 pairs consist of a black and a red card.

```
#let's simulate the experiment
N <- 10000 #number of simulated events
deck <- c(replicate(26, 0), replicate(26, 1))
couples <- replicate(N, 0)

for (ie in 1:N) {
  deck <- sample(deck, 52)
  for(ic in seq(1,51, by=2)){
    if(deck[ic]==deck[ic+1])
      couples[ie]=couples[ie]+1
  }
}

writeLines(sprintf(
  "
  Simulating %i events:
  - The mean is %.2f;
  - The standard deviation is %.2f.

  Using the Chebyshev's inequality, an upper bound of P(x<11) is: %.2f
",
  N, mean(couples), sd(couples), var(couples)/(10-mean(couples))^2
))
```

```
##
## Simulating 10000 events:
## - The mean is 13.24;
## - The standard deviation is 2.60.
##
## Using the Chebyshev's inequality, an upper bound of P(x<11) is: 0.65
```

## Exercise 6

- In a stationary bus at the departure station, a passenger gets on the bus, on average every 30 seconds.
  - Compute the probability of getting more than 6 passenger after 2 minutes. Evaluate the probability of having less than 4 passenger after 3 minutes.
  - Simulate the distribution of the arrival time of the third passenger and superimpose the corresponding pdf.
  - Repeat the procedure of the point b) for the difference in arrival time between the fifth and the first passenger.

```
#Observations
rate <- 1/30. #passengers/second

#The number of passenger is a random value following a Poisson distribution.
p.more6 <- 1 - ppois(6, lambda = rate*120)
p.less4 <- ppois(3, lambda = rate*180)

writeLines(sprintf(
  "
  The probability of getting more than 6 passenger after 2 minutes is: %.2f;
  The probability of having less than 4 passenger after 3 minutes is: %.2f.
",
  p.more6, p.less4
))
```

```
##
## The probability of getting more than 6 passenger after 2 minutes is: 0.11;
## The probability of having less than 4 passenger after 3 minutes is: 0.15.
```

```
N <- 10000 #number of simulated events

#let's simulate N times the waiting times between the arrival of three passengers
three_passengers_dt <- matrix(rexp(3 * N, rate), nrow = N, ncol = 3)
third_arrival_times <- rowSums(three_passengers_dt)

max_t <- max(third_arrival_times)

x_plot <- seq(from=0, to=max_t*1.1, length.out=100)
y_plot <- dgamma(x_plot, shape = 3, rate = rate)

hist(third_arrival_times, freq = FALSE, col = color_vector[5], xlab = "t [s]", ylab = "Density", main = "
lines(x_plot, y_plot, type='l', lwd = 2, col = color_vector[1], lty = linetype[1])
polygon(x_plot, y_plot, col = adjustcolor(color_vector[1], alpha.f = 0.1))

legend_labels <- paste("Expected Erlang distribution")
legend("topright", legend = legend_labels, col = color_vector[1], lty = linetype[1], lwd = 2)
```



```
N <- 10000 #number of simulated events

five_passengers_dt <- matrix(rexp(5 * N, rate), nrow = N, ncol = 5)
first_arrival_times <- five_passengers_dt[,1]
fifth_arrival_times <- rowSums(five_passengers_dt)

first_fifth_time <- fifth_arrival_times - first_arrival_times

max_t <- max(first_fifth_time)

#from a theoretical POV, I can just consider the waiting time of the 4th passenger
x_plot <- seq(from=0, to=max_t*1.1, length.out=100)
y_plot <- dgamma(x_plot, shape = 4, rate = rate)

hist(first_fifth_time, freq = FALSE, col = color_vector[5], xlab = "t [s]", ylab = "Density", main = "
lines(x_plot, y_plot, type='l', lwd = 2, col = color_vector[1], lty = linetype[1])
polygon(x_plot, y_plot, col = adjustcolor(color_vector[1], alpha.f = 0.1))

legend_labels <- paste("Expected Erlang distribution")
legend("topright", legend = legend_labels, col = color_vector[1], lty = linetype[1], lwd = 2)
```

