Exercícios Gerais

Thierry Martins Ribeiro

08/08/2025

```r
library(stringr)
```

link dos exercícios:

```r
url <- "https://livro.curso-r.com/7-4-o-pacote-stringr.html#exerc%C3%ADcios-19 "
```

PARTE 1

4. Imagine que a seguinte string é a parte final de uma URL.

“/ac/rio-branco/xpto-xyz-1-0-1fds2396-5.``` Transforme-a em "AC - Rio Branco" utilizando funções do pacote {stringr}.

```r
url <- c('/ac/rio-branco/xpto-xyz-1-0-1fds2396-5')

partes <- str_split(url, "/", simplify = TRUE)
estado <- str_to_upper(partes[2])
cidade <- str_replace_all(partes[3], "-", " ")
cidade <- str_to_title(cidade)
resultado <- str_c(estado, " - ", cidade)

print(resultado)
```

```
## [1] "AC - Rio Branco"
```

6. De acordo com as regras da língua portuguesa, antes de “p” ou “b” devemos usar a letra “m”. Em outras palavras, com outras consoantes, usamos a letra “N”. Suponha que você tem o seguinte texto com erros gramaticais:

```r
texto <- 'Nós chamamos os bonbeiros quando começou o incêmdio.'
srt_bombeiros <- str_replace_all(texto, "bonbeiros", "bombeiros")
srt_incendio <- str_replace_all(srt_bombeiros, "incêmdio", "incêndio")
print(srt_incendio)
```

```
## [1] "Nós chamamos os bombeiros quando começou o incêndio."
```

7. Considere o seguinte texto

"A função mais importante para leitura de dados no `lubridate` é a `ymd`. Essa função serve para ler qualquer data de uma `string` no formato `YYYY-MM-DD`. Essa função é útil pois funciona com qualquer separador entre os elementos da data e também porque temos uma função para cada formato (`mdy`, `dmy`, `dym`, `myd`, `ydm`).Extraia todas as combinações da função ymd, sem repetições.

```r
texto <- "A função mais importante para leitura de dados no `lubridate` é a `ymd`. Essa função serve pa

funcoes <- str_extract_all(texto, "`[a-z]{3}`")[[1]]
funcoes_unicas <- unique(funcoes)
print(funcoes_unicas)
```

```
## [1] "`ymd`" "`mdy`" "`dmy`" "`dym`" "`myd`" "`ydm`"
```

8. Considere as frases abaixo

Crie uma regra para identificar se o texto refere-se a um feedback positivo ou negativo sobre o produto (considere "não bom = ruim" e "não ruim = bom"). Retorne um vetor lógico que vale TRUE se o feedback é positivo e FALSE caso contrário.

```r
s <- c(
  'O produto é muito bom.', #TRUE
  'O produto não é bom.', #FALSE
  'O produto não é muito bom.', #FALSE
  'O produto não é ruim.', #TRUE
  'O produto não é não bom.' #FALSE
)

feedback <- str_detect(s, "não") & str_detect(s, "bom")
feedback_positivo <- !feedback
print(feedback_positivo)
```

```
## [1]  TRUE FALSE FALSE  TRUE FALSE
```

PARTE 2

Link para os exercícios:

```
url <-"http://cursos.leg.ufpr.br/ecr/probabilidade-e-vari%C3%A1veis-aleat%C3%B3rias.html#exerc%C3%ADcio
```

#————————————————-Exercicio 1————————————————-#

Dados

```
X ~ N(90,100)
90 = MEDIA
100 = VARIANCIA # V^2 = 100 , Variancia = 10
```

a. P(X <= 115)

```
solucao_a <- pnorm(115, mean = 90, sd = sqrt(100))
print(solucao_a)
```

```
## [1] 0.9937903
```

b. P(X >= 80)

```
solucao_b <- pnorm(80, mean = 90, sd = sqrt(100))
solucao_b <- 1 - solucao_b
print(solucao_b)
```

```
## [1] 0.8413447
```

c. P(X <= 75)

```
solucao_c <- pnorm(75, mean = 90, sd = sqrt(100))
print(solucao_c)
```

```
## [1] 0.0668072
```

d. P(85 <= X <= 110)

```
solucao_d1 <- pnorm(110, mean = 90, sd = sqrt(100))
solucao_d2 <- pnorm(85, mean = 90, sd = sqrt(100))
solucao_d <- solucao_d1 - solucao_d2
print(solucao_d)
```

```
## [1] 0.6687123
```

e. P(|X-90| <= 10)

|X-90| <= 10

1° X - 90 <= 10 => X <= 100

2° -(X - 90) <= 10, -X + 90 <= 10, -X <= -80, X >= 80

```
solucao_e1 <- pnorm(100, mean = 90, sd = sqrt(100))
solucao_e2 <- pnorm(80, mean = 90, sd = sqrt(100))
solucao_e <- solucao_e1 - solucao_e2
print(solucao_e)
```

```
## [1] 0.6826895
```

f. O valor de alfa tal que $P(90-alfa <= X <= 90+alfa) = gama, gamma=0.95

```
gama <- 0.95
alfa <- qnorm((1 + gama) / 2, mean = 90, sd = sqrt(100)) - 90
print(alfa)
```

```
## [1] 19.59964
```

#————————————————–Exercicio 2————————————————–#

Sendo X uma variável seguindo o modelo binomial com parâmetros n =15 e p=0.4

Dadso

```r
n <- 15
p <- 0.4
```

a. P(X >= 14)

```r
p_a <- pbinom(13, size = n, prob = p)
p_aq <- 1 - p_a
cat("Questão a) \n")
```

```
## Questão a)
```

```r
print(p_a)
```

```
## [1] 0.9999748
```

b. P(8 < X <= 10)

```r
p_b1 <- pbinom(10, size = n, prob = p)
p_b2 <- pbinom(8, size = n, prob = p)
p_b <- p_b1 - p_b2
cat("Questão b) \n")
```

```
## Questão b)
```

```r
print(p_b)
```

```
## [1] 0.08569975
```

c. P(X < 2 ou X >= 11)

```r
p_c1 <- pbinom(1, size = n, prob = p)
p_c2 <- pbinom(10, size = n, prob = p)
p_total_c <- p_c1 + (1-p_c2)
cat("Questão c) \n")
```

```
## Questão c)
```

```r
print(p_total_c)
```

```
## [1] 0.0145197
```

d. $P(X >= 11 \text{ ou } X > 13)$

```r
p_total_d <- 1 - pbinom(10, size = n, prob = p)
cat("Questão d) \n")
```

```
## Questão d)
```

```r
print(p_total_d)
```

```
## [1] 0.009347661
```

e. $P(X > 3 \text{ ou } X < 6)$

```r
p_e1 <- pbinom(3, size = n, prob = p)
p_e1_total <- 1 - p_e1

p_e2 <- pbinom(5, size = n, prob = p)
p_intervalo <- p_e2 - p_e1

p_total_e2 <- p_e1_total + p_e2 - p_intervalo
cat("Questaõ e)\n")
```

```
## Questaõ e)
```

```r
print(p_total_e2)
```

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

f. $P(X <= 13 \mid X >= 11)$

```r
p_f1 <- pbinom(10, size = n, prob = p)
p_f2 <- pbinom(13, size = n, prob = p) - p_f1
p_f1_total <- 1 - p_f1
p_total_f <- p_f2 / p_f1_total
cat("Questão f) \n")
```

```
## Questão f)
```

```r
print(p_total_f)
```

```
## [1] 0.9973006
```

#————————————————-Exercicio 3————————————————-#

Uma empresa informa que 30% de suas contas a receber de outras empresas encontram-se vencidas. Se o contador da empresa seleciona aleatoriamente 5 contas, determine a probabilidade de:

Dados

```
n = 5
p = 0.3
```

a. Nenhuma conta vencida

```
p_a <- dbinom(0, size = n, prob = p)
cat("Questão a) \n")
```

```
## Questão a)
```

```
print(p_a)
```

```
## [1] 0.16807
```

b. Exatamente duas cotas vencidas

```
p_b <- dbinom(2, size = n, prob = p)
cat("Questão b) \n")
```

```
## Questão b)
```

```
print(p_b)
```

```
## [1] 0.3087
```

c. Três ou mais contas estarem vencidas

```
p_c <- 1 - pbinom(2, size = n, prob = p)
cat("Questão c) \n")
```

```
## Questão c)
```

```r
print(p_c)
```

```
## [1] 0.16308
```

#————————————————-Exercicio 4————————————————-#

Uma empresa recebe 720 emails em um intervalo de 8 horas. Qual a probabilidade de que :

Dado

```r
lambda <- (720 / 480)
```

a. Em 6 minutos receba pelo menos 3 emails

```r
lambda * 6  # Taxa de emails por 6 minutos
```

```
## [1] 9
```

```r
p_a <- 1 - ppois(2, lambda = lambda)
cat("Questão a) \n")
```

```
## Questão a)
```

```r
print(p_a)
```

```
## [1] 0.1911532
```

b. Em 4 minutos não receba nenhum email

```r
p_b <- dpois(0, lambda = lambda * 4)
cat("Questão b) \n")
```

```
## Questão b)
```

```r
print(p_b)
```

```
## [1] 0.002478752
```

#————————————————-Exercicio 5————————————————-#

O processo de empacotamento de uma fábrica de cereais foi ajustado de maneira que uma média de = 13 kg de cereal seja colocado em cada caixa. Sabe-se que existe uma pequena variabilidade no enchimento dos pacotes devido à fatores aleatórios, e que o desvio-padrão do peso de enchimento é de = 1kg. Assume-se que a distribuição dos pesos tem distribuição normal. Com isso, determine as probabilidades de que uma caixa escolhida ao acaso:

Dados

```r
u = 13
sigma = 1
```

a. Pese entre 13,0 e 13,2 kg.

```r
p_a <- pnorm(13.2, mean = u, sd = sigma) - pnorm(13, mean = u, sd = sigma)
cat("Questão a) \n")
```

```
## Questão a)
```

```r
print(p_a)
```

```
## [1] 0.07925971
```

b. Tenha um peso maior do que 13,25 kg

```r
p_b <- 1 - pnorm(13.25, mean = u, sd = sigma)
cat("Questão b) \n")
```

```
## Questão b)
```

```r
print(p_b)
```

```
## [1] 0.4012937
```

c. Pese entre 12,8 e 13,1 kg

```
p_c <- pnorm(13.1, mean = u, sd = sigma) - pnorm(12.8, mean = u, sd = sigma)
cat("Questão c) \n")
```

```
## Questão c)
```

```
print(p_c)
```

```
## [1] 0.1190875
```

d. Pese entre 13,1 e 13,2 kg.

```
p_d <- pnorm(13.2, mean = u, sd = sigma) - pnorm(13.1, mean = u, sd = sigma)
cat("Questão d) \n")
```

```
## Questão d)
```

```
print(p_d)
```

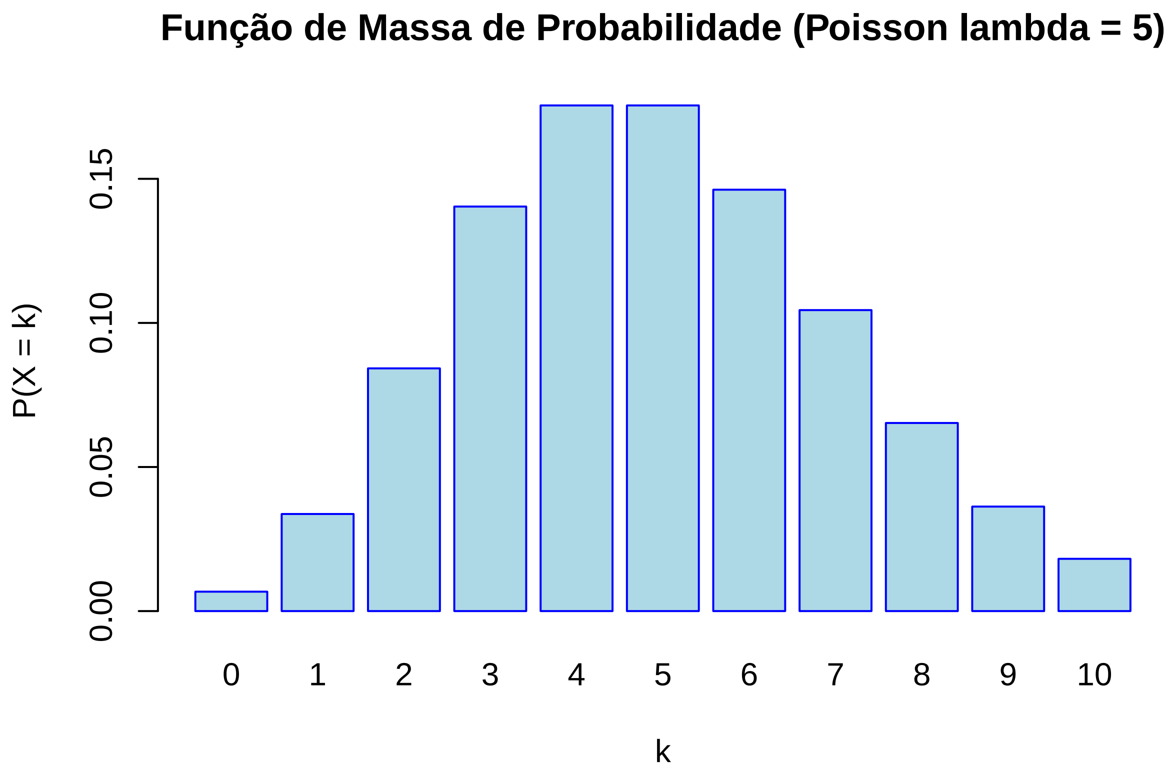
```
## [1] 0.03943187
```

#————————————————-Exercicio 6————————————————-#

Faça os seguintes gráficos

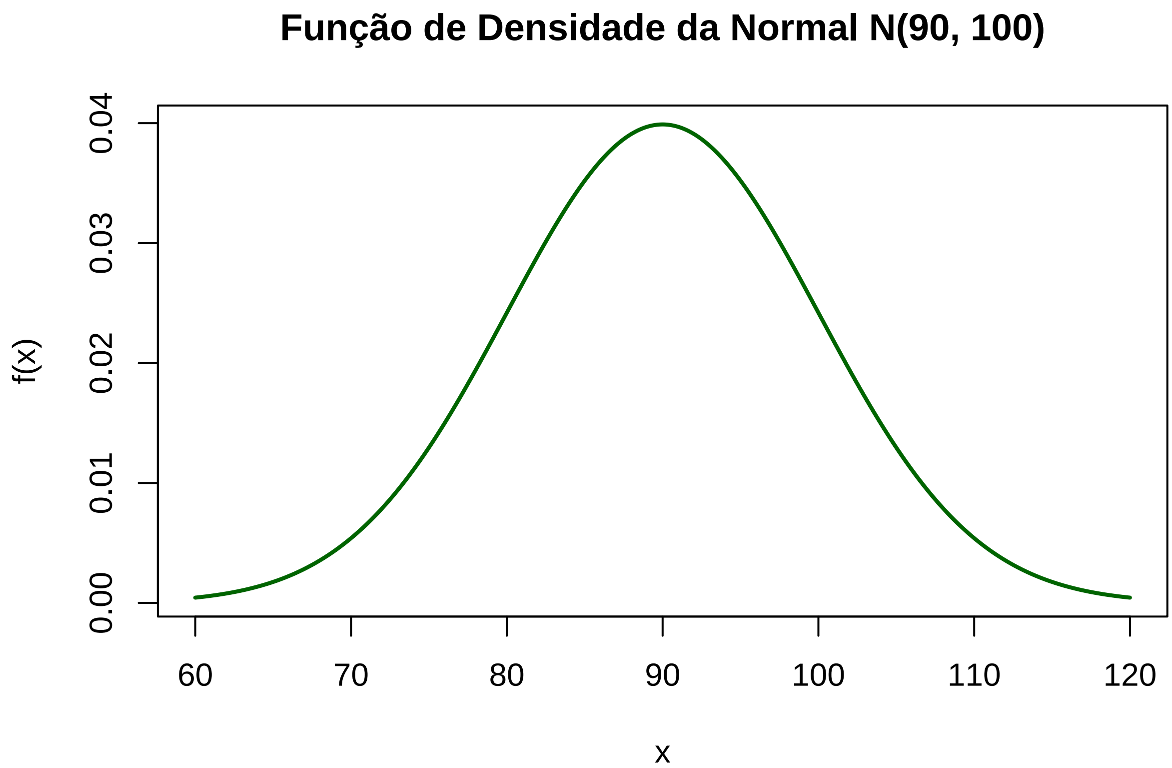
a. da função de densidade de uma variável com distribuição de Poisson com parâmetro $\lambda = 5$

```
lambda <- 5
x <- 0:10

massa <- dpois(x, lambda)

barplot(massa, names.arg = x,
        main = "Função de Massa de Probabilidade (Poisson lambda = 5)",
        xlab = "k", ylab = "P(X = k)",
        col = "lightblue", border = "blue")
```



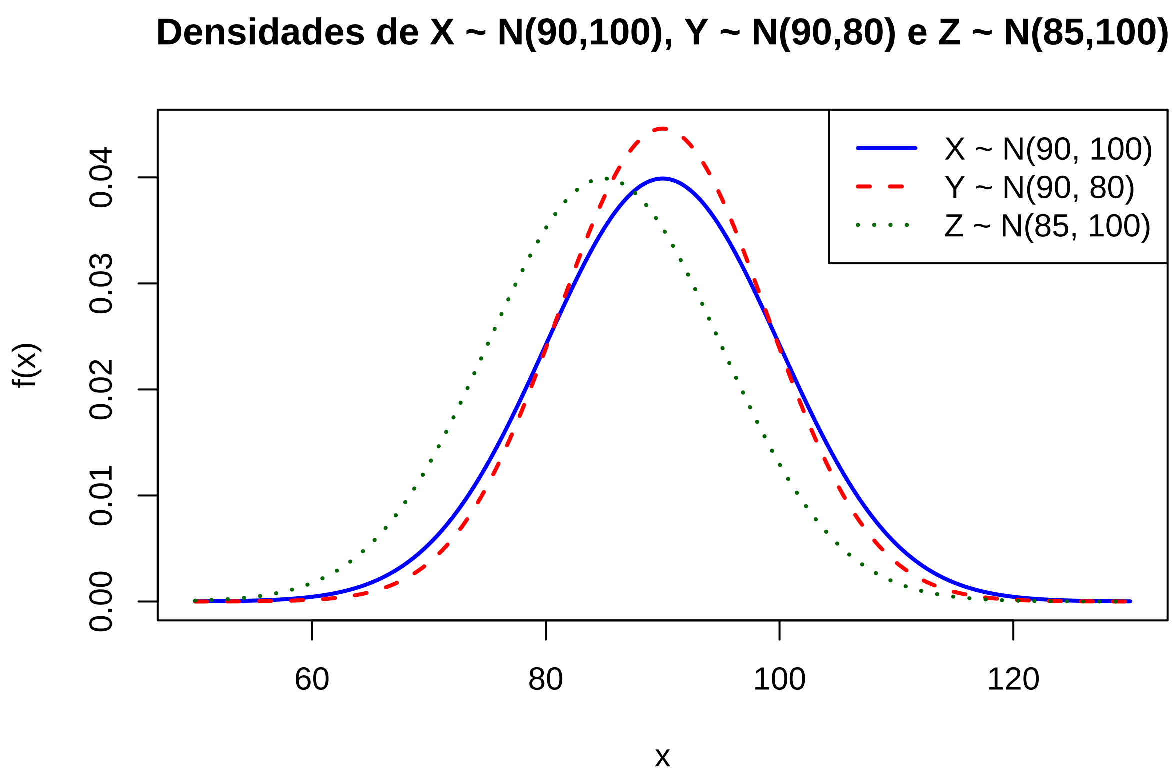
b. da densidade de uma variável X N(90,100).

```
media <- 90
desvio <- 10

x <- seq(60, 120, by = 0.1)

densidade <- dnorm(x, mean = media, sd = desvio)

plot(x, densidade, type = "l", lwd = 2,
     main = "Função de Densidade da Normal N(90, 100)",
     xlab = "x", ylab = "f(x)",
     col = "darkgreen")
```



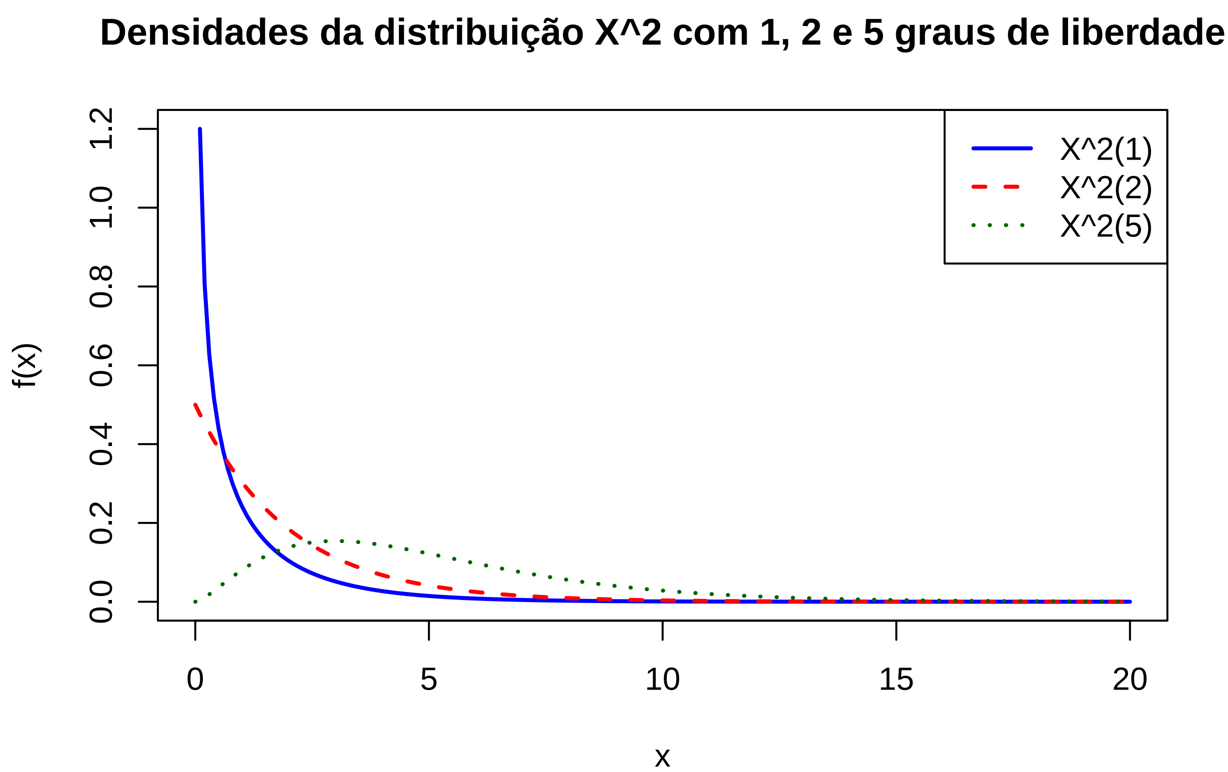
c. sobreponha ao gráfico anterior a densidade de uma variável Y N(90,80) e outra Z N(85,100).

```
x <- seq(50, 130, by = 0.1)
densidade_X <- dnorm(x, mean = 90, sd = sqrt(100))
densidade_Y <- dnorm(x, mean = 90, sd = sqrt(80))
densidade_Z <- dnorm(x, mean = 85, sd = sqrt(100))

plot(x, densidade_X, type = "l", lwd = 2, col = "blue",
     ylim = c(0, max(densidade_X, densidade_Y, densidade_Z)),
     main = "Densidades de X ~ N(90,100), Y ~ N(90,80) e Z ~ N(85,100)",
     xlab = "x", ylab = "f(x)")

lines(x, densidade_Y, col = "red", lwd = 2, lty = 2)
lines(x, densidade_Z, col = "darkgreen", lwd = 2, lty = 3)
legend("topright", legend = c("X ~ N(90, 100)", "Y ~ N(90, 80)", "Z ~ N(85, 100)"),

       col = c("blue", "red", "darkgreen"), lwd = 2, lty = c(1, 2, 3))
```



d. densidades de distribuições χ^2 com 1, 2 e 5 graus de liberdade.

```r
x <- seq(0, 20, by = 0.1)

dens_1 <- dchisq(x, df = 1)
dens_2 <- dchisq(x, df = 2)
dens_5 <- dchisq(x, df = 5)


all_dens <- c(dens_1, dens_2, dens_5)
all_dens <- all_dens[is.finite(all_dens)]

plot(x, dens_1, type = "l", lwd = 2, col = "blue",
     ylim = c(0, max(all_dens)),
     main = "Densidades da distribuição X^2 com 1, 2 e 5 graus de liberdade",
     xlab = "x", ylab = "f(x)")


lines(x, dens_2, col = "red", lwd = 2, lty = 2)
lines(x, dens_5, col = "darkgreen", lwd = 2, lty = 3)

legend("topright", legend = c("X^2(1)", "X^2(2)", "X^2(5)"),
       col = c("blue", "red", "darkgreen"), lwd = 2, lty = c(1, 2, 3))
```



Parte 3

Implemente o gerador Midsquare idealizado pelo matemático John von Neumann. Por que o gerador Midsquare não é um bom gerador? Explique

Implemente o gerador Midsquare idealizado pelo matemático John von Neumann.

- A ideia de início era gerar um n manualmente, mas o rmd nao permite isso.

```
gerador_midsquare <- function(semente, n = 10) {

  #n <- as.integer(readline(prompt = "Quantos números aleatórios deseja gerar? "))

  numeros_novos <- numeric(n)

  for (i in 1:n) {
    semente <- semente^2

    semente_str <- sprintf("%08d", semente)

    semente <- as.integer(substr(semente_str, 3, 6))
```

```r
    numeros_novos[i] <- semente
  }

  return(numeros_novos)
}



semente_inicial <- sample(1000:9999, 1)
numeros_gerados <- gerador_midsquare(semente_inicial, n = 10)
print(numeros_gerados)
```

```
##  [1] 7710 4441 7224 1861 4633 4646 5853 2576 6357 4114
```

Por que o gerador Midsquare não é um bom gerador?

- Ele nao e um bom gerador porque pode gerar sequências de números que não são uniformemente distribuídas, especialmente se a semente inicial não for escolhida adequadamente. O gerador Midsquare utiliza o quadrado da semente e extrai os dígitos do meio, o que pode levar a padrões repetitivos. Além disso, ele pode ter um período curto, o que significa que os números gerados podem se repetir rapidamente.