Statistics for Engineers
TI0111

Homework II

Nome completo: Lorena de Carvalho Gonçalves

Maria Lissa Rodrigues Costa

Numero de matricula: 567096

565974

LINK REPOSITÓRIO GITHUB

Questão 1

Em um restaurante muito frequentado, aproximadamente 70% dos clientes pedem uma sobremesa após o prato principal. Seja X a variável aleatória que representa o número de clientes que pedem sobremesa em uma amostra aleatória de $n = 50$ clientes.

1. Determine a função de distribuição de X.
2. Construa os gráficos da função massa de probabilidade (PMF) e da função distribuição acumulada (CDF) de X.
3. Calcule o valor esperado, a variância e o desvio padrão de X.
4. Calcule a probabilidade de:
   (a) $P(X \geq 20)$.
   (b) $P(30 < X < 43)$.
   (c) $P(X = 31)$.
5. Suponha que o restaurante estoque sobremesas com base na demanda esperada. Como o uso da distribuição de X poderia ajudar a reduzir desperdício e evitar falta de produtos.
6. Como mudanças em p (por exemplo, sobremesa se torna mais popular, $p = 0.8$) ou em n (número de clientes) afetariam a forma e as probabilidades de X?

Solução da questão 1

1: A função de distribuição ideal para esse caso é a distribuição binomial, dado que temos uma probabilidade fixa (p = 0.70) e um número fixo de ensaios independentes (n = 50). Logo, a variável aleatória X pode ser representada por X∼Bin(50,0.7). Dessa forma, sua função de distribuição (PMF) é representada como:

$$P(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

2: CDF calcula a probabilidade de X assumir um valor menor ou igual a um ponto específico. É representado por:

$$F(k) = P(X \leq k) = \sum_{i=0}^{k} \binom{n}{i} p^i (1-p)^{n-i}$$

Como já sabemos a PMF e CDF, podemos fazer o plot de ambas as funções:

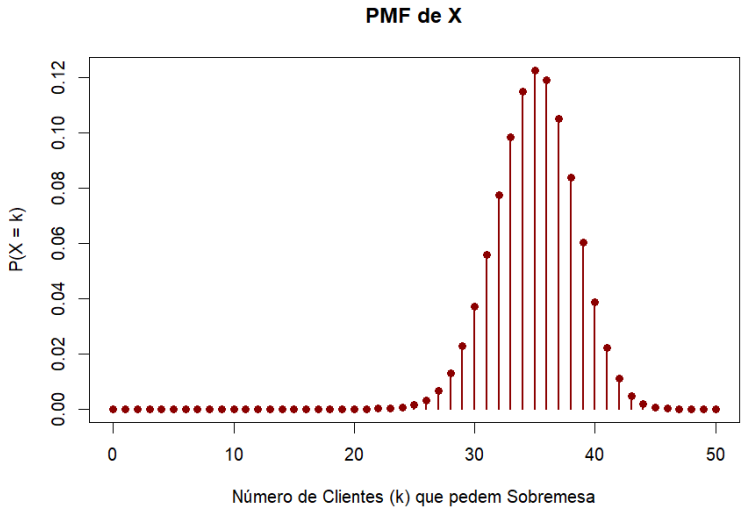

Figura 1: Gráfico PMF.

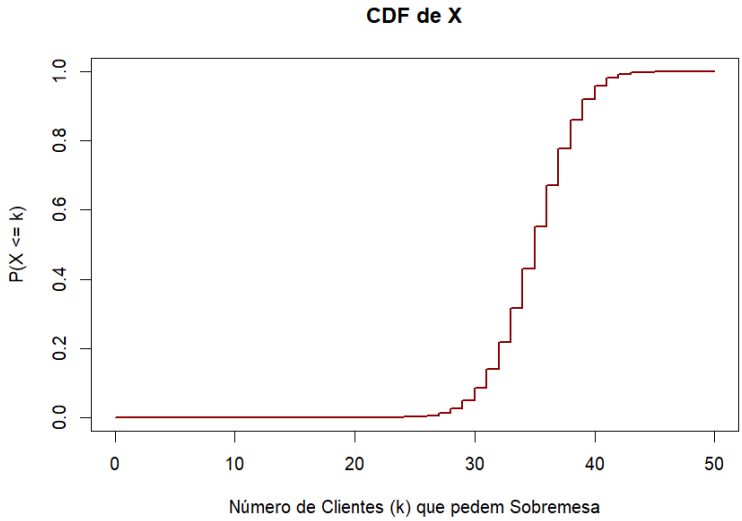

Figura 2: Gráfico CDF.

3: Calcularemos esses valores de acordo com as fórmulas vistas em sala de aula:

-Valor esperado:

$$E[X] = n \cdot p = 50 \cdot 0.7 = 35$$

-Variância:

$$Var(X) = n \cdot p \cdot (1-p) = 50 \cdot 0.7 \cdot 0.3 = 10.5$$

-Desvio padrão:

$$\sigma = \sqrt{10.5} \approx 3.2404$$

4: (a) Podemos achar essa probabilidade por meio do complemento dela, ou seja, vamos calcular a probabilidade de X ser menor que 20, por meio da somatória de probabilidades que vão de 0 até 19, e diminuir esse valor de 1:

$$P(X \geq 20) = 1 - P(X < 20) = 1 - \sum_{i=0}^{19} \binom{50}{i} (0.7)^i (0.3)^{50-i} \approx 0.999997$$

(b) Calculamos a probabilidade de X indo de 0 até 42 menos a probabilidade de X ir de 0 até 30:

$$P(30 < X < 43) = P(X \leq 42) - P(X \leq 30) = \sum_{i=31}^{42} \binom{50}{i} (0.7)^i (0.3)^{50-i} \approx 0.90793$$

(c) Apenas acrescentamos o valor X = 32 na fórmula do PDF introduzida no item 1:

$$P(X = 31) = \binom{50}{31}(0.7)^{31}(0.3)^{19} \approx 0.052739$$

5: Pois, por meio na distribuição, podemos analisar o valor de X, o número de clientes que pedem sobremesa, que tem maior probabilidade. Assim, com esse conhecimento, o restaurante pode ter um valor estimado de clientes que pedem sobremesa, o que ajuda a criar um estoque semelhante com esse valor. Dessa forma, excessos e faltas de sobremesas serão evitadas.

6: Se p aumenta para 0.8, a curva se desloca para a direita, a variância e valor esperado mudam:

$$E[X]_{novo} = 50 \cdot 0.8 = 40 > E[X]_{antigo} = 35$$

$$Var(X)_{p=0.8} = 50 \cdot 0.8 \cdot 0.2 = 8.0 < Var(X)_{p=0.7} = 10.5$$

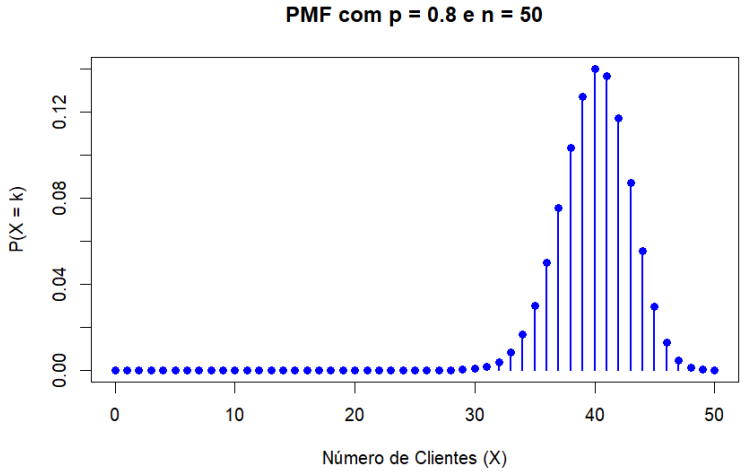

Figura 3: Curva se deslocando para direita com mudança de parâmetro p.

Se n aumenta, a distribuição torna-se mais simétrica, visto que a probabilidade total está distribuída entre um número maior de resultados possíveis.

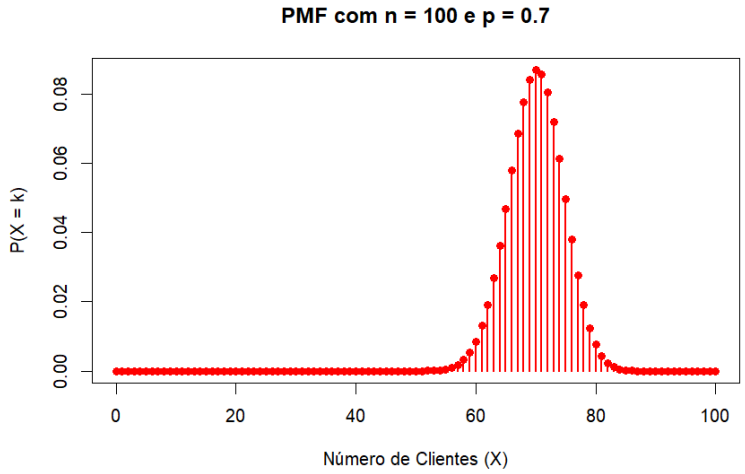

Figura 4: Curva se tornando mais simétrica com aumento de n.

Listado 1: Códiigo desenvolvido na solução da questão 1.

```
#item 2
range <- 0:50
p <- 0.7
n <- 50

pmf <- dbinom(range, size = n, prob = p)
cdf <- pbinom(range, size = n, prob = p)

plot(range, pmf, type = "h",
     main = "PMF de X",
     xlab = "N mero de Clientes (k) que pedem Sobremesa",
     ylab = "P(X = k)",
     col = "darkred",
     lwd = 2)
points(range, pmf, pch = 16, col = "darkred") # Adiciona pontos para
    clareza

plot(range, cdf, type = "s", # 's' para gr fico de escada (fun   o
    passo)
     main = "CDF de X",
     xlab = "N mero de Clientes (k) que pedem Sobremesa",
     ylab = "P(X <= k)",
     col = "darkred",
     lwd = 2)# Adiciona pontos no final de cada passo


#item 3
media <- n * p
cat("m dia:", media, "\n")
variancia <- n * p * (1 - p)
cat("vari ncia:", variancia, "\n")
```

```r
desvio_padrao <- sqrt(variancia)
cat("desvio padr o:", desvio_padrao, "\n")


#item 4
pa <- 1 - pbinom(19, size = n, prob = p)
cat("P(X >= 20) =", pa, "\n")
pb <- pbinom(42, size = n, prob = p) - pbinom(30, size = n, prob = p)
cat("P(30 < X < 43) =", pb, "\n")
pc <- dbinom(31, size = n, prob = p)
cat("P(X = 31) =", pc, "\n")


#item 6
plot(0:50, dbinom(0:50, size = 50, prob = 0.8), type = "h",
     col = "blue", lwd = 2,
     main = "PMF com p = 0.8 e n = 50",
     xlab = "N mero de Clientes (X)", ylab = "P(X = k)")
points(0:50, dbinom(0:50, size = 50, prob = 0.8), pch = 16, col = "blue")

plot(0:100, dbinom(0:100, size = 100, prob = 0.7), type = "h",
     col = "red", lwd = 2,
     main = "PMF com n = 100 e p = 0.7",
     xlab = "N mero de Clientes (X)", ylab = "P(X = k)")
points(0:100, dbinom(0:100, size = 100, prob = 0.7), pch = 16, col = "red
    ")
```

Questão 2

Um site realiza uma pesquisa online e oferece uma recompensa a um usuário selecionado aleatoriamente que responde a uma série de perguntas. Cada um dos 10 milhões de visitantes diários tem, independentemente, probabilidade $p = 10^{-7}$ de ganhar a recompensa.

1. Encontre uma aproximação simples e adequada para a função de massa de probabilidade (PMF) do número de vencedores em um dia, X. Justifique claramente se essa aproximação é apropriada para os valores dados de n and p.

2. Calcule o valor esperado, $E[X]$, e a variância $\mathrm{Var}(X)$, usando tanto a distribuição exata quanto a aproximada. Comente sobre a semelhança entre os resultados.

3. Suponha que você ganhe a recompensa, mas que possa haver outros vencedores. Seja $W \sim \mathrm{Pois}(1)$ o número de vencedores além de você. Se houver vários vencedores, o prêmio é sorteado aleatoriamente entre todos eles. Encontre a probabilidade de que você realmente receba o prêmio.

4. Gere um grande número de simulações diárias para o número de vencedores. Crie uma comparação visual entre os resultados empíricos e a aproximação considerada no item 1. Descreva brevemente o que a visualização indica sobre a qualidade da aproximação.

Solução da questão 2

1: Para os valores de n = 10.000.000 (tentativas ao dia) e p = 10^{-7} (chance de sucesso para cada tentativa), a distribuição mais evidente para esse problema seria X∼Bin(10^7, 10^{-7}), para X uma variável aleatória que representa o número de vencedores por dia. No entanto, para simplificar os valores utilizados, podemos pensar em uma distribuição de Poisson com taxa λ diario de 10^7 x 10^{-7}, ou seja, X∼Poiss(1).

Essa aproximação se mostra apropriada devido ao dado n = 10^7 ser uma estimativa de visitantes por dia, e o parâmetro λ da distribuição de Poisson representar o número médio esperado de eventos em determinado intervalo temporal. Nesse caso, λ = n x p = 10^7 x 10^{-7} = 1.

Com essa aproximação, podemos escrever a PMF como:

$$PMF(X = k) = \frac{e^{-1}}{k!},$$

para k = 0, 1, 2, 3...

2: Para a distribuição exata, X∼Bin(10^7, 10^{-7}), temos que o valor esperado e a variância da distribuição se comportam do seguinte modo:

$$E[X] = n * p$$

$$Var(X) = n * p * (1 - p)$$

Logo, temos:

$$E[X] = 10^7 * 10^{-7} = 1$$

$$Var(X) = 10^7 * 10^{-7} * (1 - 10^{-7}) = 0.9999999$$

Para a distribuição aproximada, X∼Poiss(1), temos que o valor esperado e a variância da distribuição de Poisson são:

$$E[X] = \lambda$$

$$Var(X) = \lambda$$

Logo, temos:

$$E[X] = \lambda = 1$$

$$Var(X) = \lambda = 1$$

Podemos inferir, a partir dos valores esperado e de variância, que a aproximação de Poisson é uma boa alternativa para simplificar a distribuição binomial de n muito grande e p muito pequeno, já que a diferença entre os valores observados é tão pequena que se torna desprezível. Logo, torna-se claro que a aproximação funciona.

3: Chamando de W a variável aleatória que representa o número de vencedores além do vencedor confirmado. Como o prêmio será sorteado aleatoriamente entre o total de vencedores, sabemos que a chance de que realmente o prêmio seja recebido é dada por $\frac{1}{nvencedores}$. Logo, ela se dá por:

$$\frac{1}{W+1},$$

em que W é W$\sim$Poiss(1).

Como o valor da variável W não tem um valor fixo, precisamos trabalhar com o valor esperado da probabilidade. Logo, teremos:

$$P = E[\frac{1}{W+1}]$$

Podemos calcular o valor esperado por meio da seguinte equação:

$$E[X] = \sum_{k=0}^{+\infty} P(X = x) * x$$

$$P = \sum_{k=0}^{+\infty} \frac{1}{k+1} * PMF(W = k),$$

Desse modo, podemos desenvolver:

$$P = \sum_{k=0}^{+\infty} \frac{e^{-1}}{(k+1) * k!} = \frac{e^{-1}}{(k+1)!},$$

$$P = e^{-1} * \sum_{k=0}^{+\infty} \frac{1}{(k+1)!}$$

Sabemos que

$$\sum_{k=0}^{+\infty} \frac{1}{k!} = e$$

Logo, como no somatório que estamos observando, o valor de $\frac{1}{0!} = 1$ não é contemplado, podemos afirmar que

$$\sum_{k=0}^{+\infty} \frac{1}{(k+1)!} = e - 1$$

Desse modo,

$$P = e^{-1} * \sum_{k=0}^{+\infty} \frac{1}{(k+1)!} = e^{-1}(e-1) = 1 - e^{-1}$$

$$P \approx 0,632$$

4: Para o item 4, geradas as simulações, tanto da distribuição binomial com n e p dados na questão, geramos um histograma para cada uma delas para melhor observá-las visualmente.

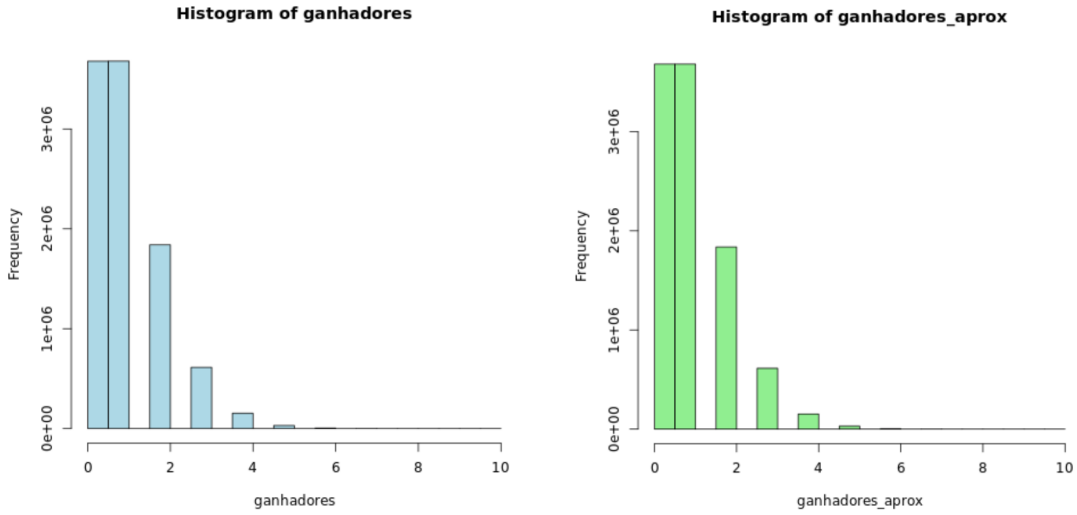

Figura 5: À esquerda, histograma da distribuição empírica, e à direita histograma da distribuição aproximada

Comparando os dois gráficos, podemos observar a semelhança evidente entre o comportamento das distribuições binomial (resultados empíricos) e a de Poisson (aproximação), o que indica que a aproximação é precisa em relação à distribuição original, pois se comporta de um jeito muito semelhante a ela.

Listado 2: Codigo desenvolvido na solução da questão 2.

```
#item 2
    #BINOMIAL
    n <- 10000000
    p <- 0.0000001
    exp <- n*p
    var <- n*p*(1-p)
    cat("Valor esperado: ", exp, "\n")
    cat("Variancia: ", exp)
    #O valor esperado e variancia da distr. Poisson = 1


#item 3
    k <- 0:40
    p <- sum(dpois(k, lambda = 1)*(1/(k+1)))
    cat("A probabilidade de ganhar o pr mio  : ", p)


#item 4
    #SIMULA AO BINOMIAL
    ganhadores <- rbinom(n = 10000000, size = 10000000, prob =
        0.0000001)
    hist(ganhadores, col= "lightblue")

    #SIMULA AO POISSON (Aprox.)
    ganhadores_aprox <- rpois(n = 10000000, lambda = 1)
    hist(ganhadores_aprox, col = "lightgreen")
```

Questão 3

Você é responsável por monitorar a temperatura de uma CPU multicore em uma unidade de processamento embarcada. Sob carga normal, a temperatura da CPU apresenta flutuações devido a mudanças na carga de trabalho, nas condições ambientais e na eficiência do sistema de resfriamento. Testes mostram que a temperatura em regime estacionário da CPU segue uma distribuição normal com temperatura média $\mu = 62\,^{\circ}\text{C}$ e desvio-padrão $\sigma = 3,5\,^{\circ}\text{C}$. Sua tarefa é simular medições de temperatura da CPU e analisar suas propriedades estatísticas.

1. Crie uma função que gere valores com distribuição normal usando a transformação de Box-Muller[1], a partir de entradas aleatórias uniformes. Especificamente:

   (a) Gere duas variáveis aleatórias uniformes independentes: $U_1, U_2 \sim \text{Unif}(0, 1)$.

   (b) Calcule dois valores normais padrão usando as fórmulas de Box-Muller:

   $$Z_1 = \sqrt{-2\ln(U_1)}\cos(2\pi U_2), \quad Z_2 = \sqrt{-2\ln(U_1)}\sin(2\pi U_2)$$

   Z_1 e Z_2 são variáveis aleatórias independentes com distribuição normal padrão. Concatene-as para formar um vetor Z de valores normais padrão.

[1] https://en.wikipedia.org/wiki/Box-Muller_transform

(c) Converta cada valor normal padrão para a distribuição de temperatura da CPU:

$$T = 62 + 3.5\,Z$$

2. Use seu gerador de números aleatórios para gerar 1.000 medições de temperatura da CPU.

   Gere mais 1.000 valores de temperatura utilizando o gerador de números aleatórios normal embutido do R, com a mesma média e desvio-padrão.

3. Para ambos os conjuntos de dados simulados, calcule:

   (a) Média amostral.

   (b) Desvio-padrão amostral.

   (c) Temperatura mínima e máxima observada.

   (d) Probabilidade empírica e teórica $P(T > 68)$.

   (e) Probabilidade empírica e teórica $P(60 < T < 65)$.

   (f) Probabilidade teórica $P(T > 75)$.

   Algum dos conjuntos de dados simulados (1.000 amostras) contém valores acima de 75°C? Caso não, explique por que eventos raros requerem tamanhos de amostra grandes para serem observados.

4. Visualize os resultados criando:

   (a) Um histograma das temperaturas simuladas da CPU (pode plotar os dois conjuntos de dados separadamente ou sobrepostos).

   (b) A função densidade de probabilidade (PDF) normal teórica (média $62\,^{\circ}\mathrm{C}$, desvio padrão $3,5\,^{\circ}\mathrm{C}$) sobreposta ao histograma.

5. Discuta seus resultados respondendo às seguintes perguntas: As distribuições empíricas da temperatura da CPU se assemelham à curva normal teórica? Quão próximas estão a média amostral e o desvio-padrão amostral dos valores esperados $62\,^{\circ}\mathrm{C}$ e $3,5\,^{\circ}\mathrm{C}$? Há diferenças perceptíveis entre o conjunto de dados gerado com seu RNG manual e o produzido pelo RNG embutido do R? Como essa simulação pode ajudar na avaliação de estratégias de resfriamento ou de escalonamento dinâmico de clock? Por que geradores de números aleatórios uniformes são a base dos sistemas de RNG?

Solução da questão 3

Nesta questão, o uso da semente (seed) é indispensável para garantir a reprodutibilidade do experimento. Como o computador utiliza algoritmos matemáticos para simular o acaso, a semente define o ponto de partida exato dessa sequência. Sem ela, os valores de média, desvio-padrão e temperaturas máximas mudariam a cada execução, tornando impossível para outra pessoa validar os resultados apresentados no relatório. Além disso, a semente assegura a consistência entre o texto e os gráficos, garantindo que as conclusões sobre eventos raros, como a probabilidade de a CPU ultrapassar 75°C, sejam baseadas em um conjunto de dados fixo e verificável. Dessa forma, colocaremos nossa semente = 42.

1: (a) A Transformação de Box-Muller é um método para gerar números aleatórios com distribuição normal a partir de uma distribuição uniforme. Logo, criaremos duas variáveis uniformes. A função densidade de probabilidade (PDF) para $U \sim \text{Unif}(0, 1)$ é:

$$f(u) = \begin{cases} 1 & \text{se } 0 \leq u \leq 1 \\ 0 & \text{caso contrário} \end{cases}$$

Como a transformação de Box-Muller gera simultaneamente dois valores normais independentes (Z_1 e Z_2) a partir de um único par de entradas uniformes, para obter um vetor final de tamanho n, geramos amostras de tamanho $k = n/2$ para U_1 e U_2. Colocamos $n = 1000$.

(b) Utilizamos as variáveis uniformes para calcular dois valores que seguem a distribuição normal padrão ($Z \sim N(0, 1)$). As fórmulas de Box-Muller transformam coordenadas cartesianas uniformes em coordenadas polares que resultam em uma distribuição gaussiana:

$$Z_1 = \sqrt{-2\ln(U_1)}\cos(2\pi U_2)$$

$$Z_2 = \sqrt{-2\ln(U_1)}\sin(2\pi U_2)$$

(c) A distribuição normal padrão (Z) possui média 0 e desvio padrão 1. Para modelar a temperatura da CPU (T) com os parâmetros específicos do problema ($\mu = 62$ e $\sigma = 3, 5$), aplicamos a seguinte transformação linear:

$$T = \mu + \sigma Z$$

$$T = 62 + 3, 5Z$$

2: Nesta etapa, utilizamos a função criada anteriormente. A lógica fundamental é que, para obter as 1.000 medições solicitadas, precisamos de 500 pares de entradas uniformes, já que cada par (U_1, U_2) produz dois valores de saída (Z_1, Z_2).

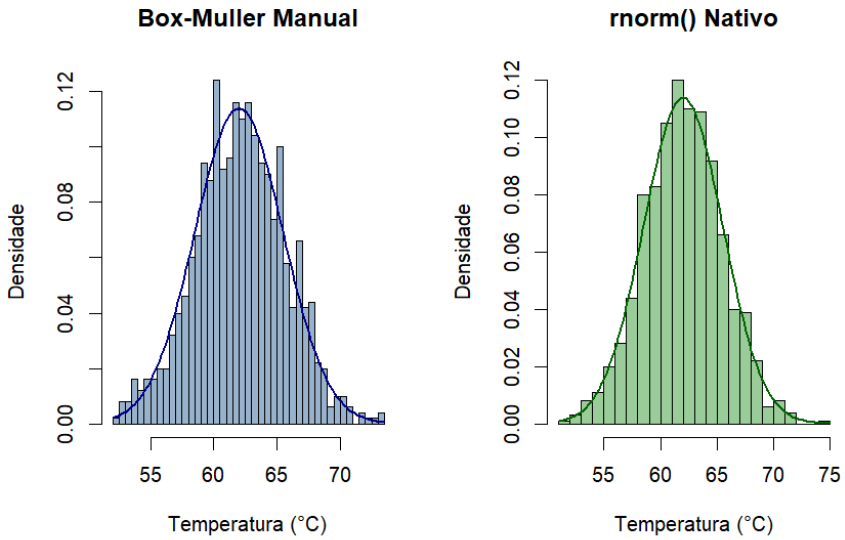

Figura 6: Comparação gerador aleatório manual (com ponto de partida setado em 42) e o gerador aleatório embutido na linguagem R.

A conclusão demonstra que os métodos de Box-Muller e rnorm são estatisticamente equivalentes, pois ambos convergem para a média de 62°C e desvio padrão de 3,5°C. Embora o histograma manual apresente flutuações amostrais naturais, ele segue a silhueta da curva teórica. A sobreposição da densidade confirma que a transformação das variáveis uniformes foi bem-sucedida, validando o gerador manual para simular com precisão o comportamento térmico da CPU.

3: (a) A média amostral dos conjuntos representa a tendência central dos dados simulados. A média pode ser calculada por meio da seguinte fórmula:

$$media = \frac{\sum_{i=1}^{N} Amostra_i}{N},$$

para N o número de amostras simuladas. Como temos um número muito grande de amostras, não conseguimos realizar esse cálculo manualmente. Na linguagem R, podemos obter esse valor por meio da função nativa do R:

Listado 3: Cálculo da média.

```
media_manual <- mean(temp_manual)
media_nativa <- mean(temp_nativa)
```

Ou então, por meio de:

Listado 4: Cálculo da média.

```
media_manual <- (sum(temp_manual))/1000
media_nativa <- (sum(temp_nativa))/1000
```

Desse modo, encontramos uma média da distribuição manual de 62.12303, e uma média da distribuição nativa de 61.98139. Esses valores estão próximos de 62.

(b) O desvio padrão amostral é uma medida que representa a dispersão dos dados de uma amostra em relação a média amostral. Ele pode ser calculado por meio da seguinte expressão:

$$D.P = \sqrt{\frac{\sum_{i=1}^{N}(Amostra_i - media)^2}{N-1}},$$

para N o número de amostras. No R, podemos calcular com a função nativa do R:

Listado 5: Cálculo do desvio padrão.

```
dp_manual <- sd(temp_manual)
dp_nativa <- sd(temp_nativa)
```

Ou então, por meio de:

Listado 6: Cálculo do desvio padrão.

```
sd_manual <- sqrt((sum((temp_manual - media_manual)^2))/(1000-1))
sd_nativa <- sqrt((sum((temp_nativa - media_nativa)^2))/(1000-1))
```

Em ambos os modos, recebemos um desvio padrão de 3.58019, para a distribuição manual, e de 3.451213 para a distribuição nativa do R. Ambos os valores são similares e estão de acordo com o valor de $\sigma = 3,5$ dado na questão.

(c) Para encontrarmos os valores de temperatura máxima e mínima dentre as distribuições, utilizaremos:

Listado 7: Temperaturas máxima e mínima.

```
max_manual <- max(temp_manual)
max_nativa <- max(temp_nativa)

min_manual <- min(temp_manual)
min_nativa <- min(temp_nativa)
```

Os valores de temperatura retornados foram:
Max amostral manual = 73.23601
Max amostral nativa = 74.54631
Min amostral manual = 52.3552
Min amostral nativa = 51.75033

(d) Queremos encontrar a probabilidade empírica (por meio das simulações) e teórica de ambas as distribuições, manual e nativa, de P(T > 68).

Inicialmente, tratando da probabilidade empírica, podemos associar esse valor à:

$$p.empirica = \frac{n(T > 68)}{N},$$

para N o número de amostras. Podemos calcular esse valor, sabendo o tamanho de N, dessa forma, na linguagem R:

Listado 8: Cálculo de P(T>68) empírico.

```
cont_manual <- 0
cont_nativa <- 0

for(i in 1:1000){
    if(temp_manual[i] > 68){
        cont_manual <- cont_manual + 1
    }
    if (temp_nativa[i] > 68){
        cont_nativa <- cont_nativa + 1
    }
}

cat("P(T>68) manual  : ", cont_manual/1000)
cat("P(T>68) nativa  : ", cont_nativa/1000)
```

A estrutura for conta a quantidade de amostras cujo valor estão acima de 68, tanto para a distribuição manual quanto para a nativa, e divide esse valor pelo número de amostras. A partir desse código, temos que P(T>68) empírica manual é = 0.043, e que P(T>68) empírica nativa = 0.041.

Em seguida, tratando da probabilidade teórica, podemos normalizar a distribuição normal de modo que ela vire a distribuição padrão (média = 0 e variância = 1), para assim encontrar valores da CDF para a distribuição padrão na tabela. Sabemos que a probabilidade para T > 68 é equivalente a 1 - P(T<68).

$$P(T > 68) = 1 - \Phi(\frac{68 - 62}{3.5}) = 1 - \Phi(1,71) = 1 - 0.95637$$

$$P(T > 68)teorico = 0,04363$$

Na linguagem R, esse valor pode ser obtido com o comando nativo:

Listado 9: Cálculo de P(T>68) teórico.

```
prob_teo_68 <- 1 - pnorm(68, mean = 62, sd = 3.5)
```

Que nos retorna o valor de 0.04323813, que é condizente com o cálculo realizado, já que o nosso valor de 1,71 foi aproximado.

(e) Queremos encontrar o valor da probabilidade de 60 < T < 65. Primeiramente, vamos encontrar o valor empírico para a distribuição manual e para a distribuição nativa. Realizaremos um processo análogo ao ítem anterior para esse cálculo:
Na linguagem R, esse valor pode ser obtido com o comando nativo:

Listado 10: Cálculo de P(60 < T < 65) empírico.

```
cont_manual <- 0
        cont_nativa <- 0

        for(i in 1:1000){
            if(temp_manual[i] > 60 & temp_manual[i] < 65){
```

```
            cont_manual <- cont_manual + 1
        }
        if (temp_nativa[i] > 60 & temp_nativa[i] < 65){
            cont_nativa <- cont_nativa + 1
        }
    }

    cat("P(60 < T < 65) manual empirica  :", cont_manual/1000,
        "\n")
    cat("P(60 < T < 65) nativa empirica  :", cont_nativa/1000,
        "\n")
```

Para esse, temos que P(60 < T < 65) manual empírica = 0.508 , e P(60 < T < 65) nativa empírica = 0.536.

Para o cálculo teórico, sabemos que podemos encontrar essa probabilidade por $P(T < 65) - P(T < 60)$ para englobar os valores pedidos. Normalizando-os, analogamente ao item anterior:

$$P(60 < T < 65) = \Phi(\frac{65 - 62}{3.5}) - \Phi(\frac{60 - 62}{3.5})$$

$$P(60 < T < 65) = \Phi(0,86) - \Phi(-0,57) = \Phi(0,86) - (1 - \Phi(0,57))$$

$$P(60 < T < 65) = 0,80511 - (1 - 0.71566) = 0,52077$$

No R:

Listado 11: Cálculo de P(60 < T < 65) teórico.

```
prob_teo_6065 <- pnorm(65, mean = 62, sd = 3.5) - pnorm(60,
    mean = 62, sd = 3.5)
```

O valor teórico recebido foi de P(60 < T < 65) = 0.5204624. Assim como o ítem anterior, uma ligeira diferença pode decorrer das aproximações feitas no nosso cálculo.

(f) Queremos calcular apenas a probabilidade teórica de P(T>75). Sabemos que essa probabilidade pode ser calculada como $1 - P(T < 75)$. Logo, temos que:

$$P(T > 75) = 1 - \Phi(\frac{75 - 62)}{3.5} = 1 - \Phi(3,71)$$

O valor para $\Phi(3,71)$ é extremamente perto de 1, aproximado para 0,9999. Logo, temos que $P(T > 75) \approx 0,0001$.

No R:

Listado 12: Cálculo de P(T>75) teórico.

```
prob_teo_75 <- 1 - pnorm(75, mean = 62, sd = 3.5)
```

Para o qual nos é retornado o valor de $P(T > 75) = 0.0001018892$. Na distribuição que realizamos, o valor máximo encontrado em ambas não chega em 75, já que recebemos os valores:

Max amostral manual = 73.23601

Max amostral nativa = 74.54631

Essa ocorrência está em consonância com a probabilidade encontrada no ítem (f), que é muito baixa e caracteriza um evento raro. Na nossa simulação, que é de 1000 amostras, é esperado que, desses eventos, $0.0001018892 * 1000 \approx 0.1$ seja maior que 75. Ou seja, para que esse valor seja mais significativo, o número de amostras precisa ser de 10000 ou mais, o que tornaria o número de amostras > 75 esperados maior que 1.

4: (a)

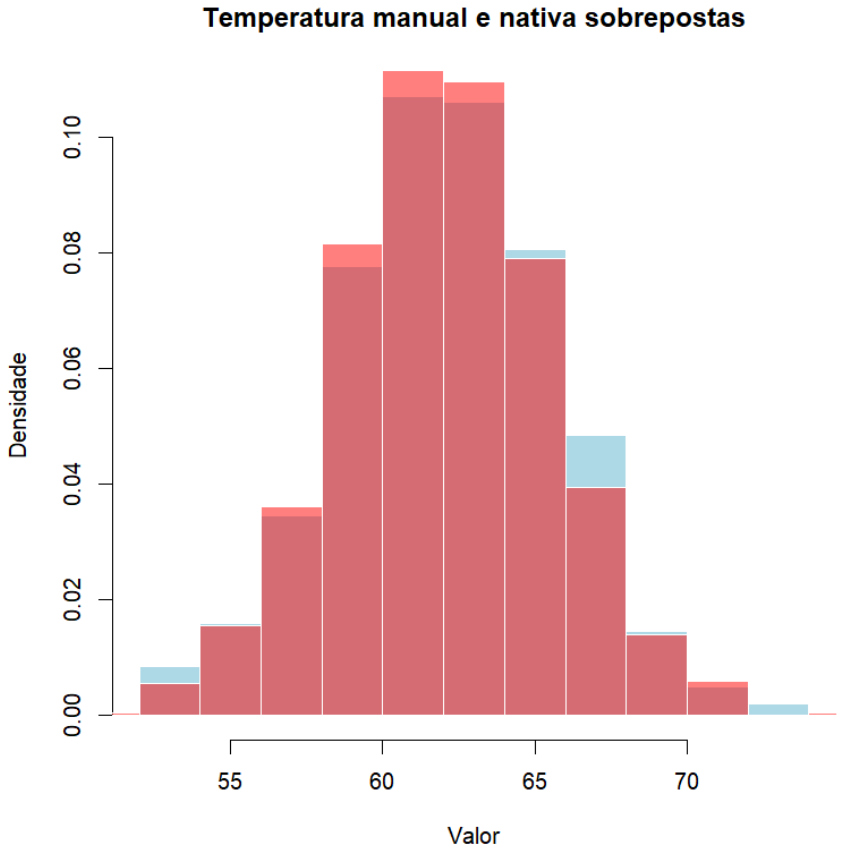

Figura 7: Temperaturas amostrais sobrepostas, manual em cor azul e nativa ao R em cor vermelha.

(b)

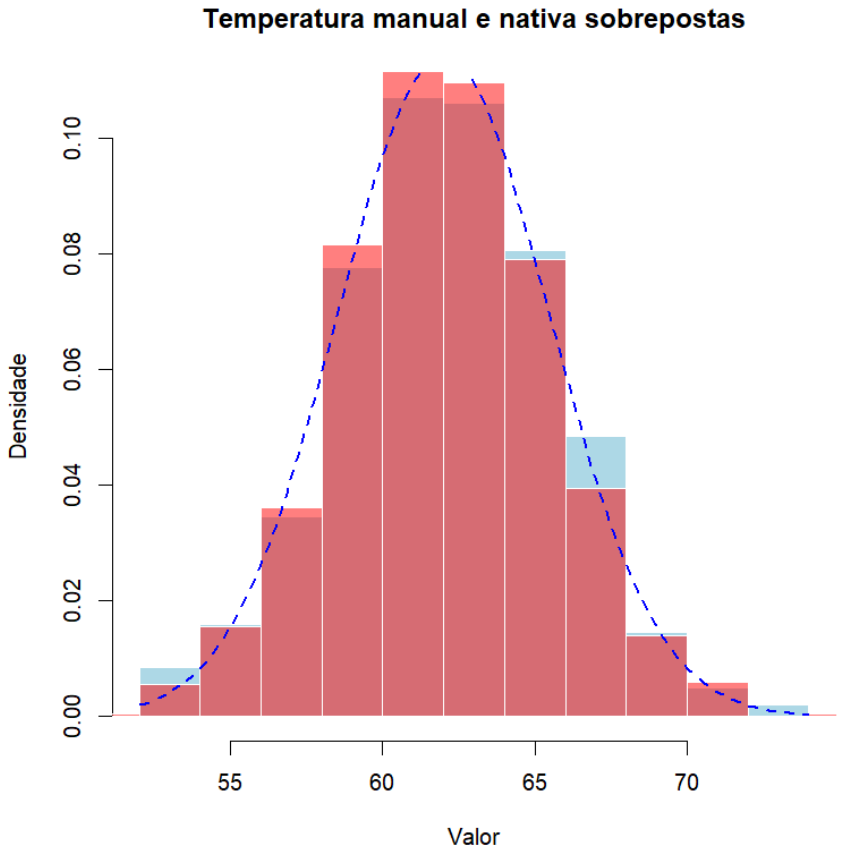

Figura 8: PDF da distribuição normal sobreposta ao histograma anterior.

5:

Observando os histogramas gerados no item anterior, é possível notar uma grande sobreposição nas temperaturas do RNG manual e no RNG rnorm(), e que ambos possuem grande semelhança com a curva da PDF da distribuição normal teórica de média 62 e desvio padrão 3.5. A distribuição segue o mesmo desenho da curva, mostrando a consonância das distribuições randomizadas geradas ao comportamento teórico.

Os valores encontrados de média amostral e desvio padrão contribuem para essa associação, já que ambas estão muito próximas do valor de $62°C$, para a média, e de $3.5°C$, para o desvio padrão.

Média amostral da distribuição manual = 62.12303
Média amostral da distribuição nativa = 61.98139
Desvio padrão da distribuição manual = 3.58019
Desvio padrão da distribuição nativa = 3.451213

Em relação aos conjuntos de dados gerados manualmente e pelo gerador embutido do R, ambos são muito similares à distribuição normal teórica, o que comprova a funcionalidade estatisticamente equivalente da transformação de Box-Muller.

As simulações utilizadas podem prever o funcionamento da CPU em condições normais de uso e, a partir da observação desses valores, verificar se o processador consegue operar com temperaturas seguras.

Por fim, os geradores de números em distribuição uniforme são base dos sistemas aleatórios pois são de simples implementação e podem ser transformados de modo a gerar variáveis em outras distribuições, como fizemos com a distribuição normal.

Listado 13: Codigo desenvolvido na solução da questão 3.

```
set.seed(42)
#item 1
n <- 500
u1 <- runif(n)
u2 <- runif(n)

z1 <- sqrt(-2 * log(u1)) * cos(2 * pi * u2)
z2 <- sqrt(-2 * log(u1)) * sin(2 * pi * u2)

z <- c(z1, z2)

temperatura_cpu <- 62 + 3.5 * z

head(temperatura_cpu)

#item2
u1 <- runif(n)
u2 <- runif(n)

z1 <- sqrt(-2 * log(u1)) * cos(2 * pi * u2)
z2 <- sqrt(-2 * log(u1)) * sin(2 * pi * u2)

temp_manual <- 62 + 3.5 * c(z1, z2)
par(mfrow = c(1, 2))
hist(temp_manual, breaks = 30, probability = TRUE,
     col = rgb(0.2, 0.4, 0.6, 0.5), main = "Box-Muller␣Manual",
     xlab = "Temperatura␣( C )", ylab = "Densidade")
curve(dnorm(x, mean = 62, sd = 3.5), add = TRUE, col = "darkblue", lwd =
    2)

temp_nativa <- rnorm(n = 1000, mean = 62, sd = 3.5)
hist(temp_nativa, breaks = 30, probability = TRUE,
     col = rgb(0.2, 0.6, 0.2, 0.5), main = "rnorm()␣Nativo",
     xlab = "Temperatura␣( C )", ylab = "Densidade")
curve(dnorm(x, mean = 62, sd = 3.5), add = TRUE, col = "darkgreen", lwd =
     2)

par(mfrow = c(1, 1))

#item3
#(a)MEDIA-AMOSTRAL
```

```r
media_manual <- mean(temp_manual)
media_nativa <- mean(temp_nativa)

cat("Media amostral manual: ", media_manual, "\n")
cat("Media amostral nativa: ", media_nativa, "\n")

#(b)DESVIO-PADRAO-AMOSTRAL
dp_manual <- sd(temp_manual)
dp_nativa <- sd(temp_nativa)

cat("D. padr o amostral manual: ", dp_manual, "\n")
cat("D. padr o amostral nativa: ", dp_nativa, "\n")

#(c)TEMP-MAX-MIN

max_manual <- max(temp_manual)
max_nativa <- max(temp_nativa)

cat("Max amostral manual: ", max_manual, "\n")
cat("Max amostral nativa: ", max_nativa, "\n")

min_manual <- min(temp_manual)
min_nativa <- min(temp_nativa)

cat("Min amostral manual: ", min_manual, "\n")
cat("Min amostral nativa: ", min_nativa, "\n")

#(d)P(T>68)
#EMPIRICA
cont_manual <- 0
cont_nativa <- 0

for(i in 1:1000){
  if(temp_manual[i] > 68){
    cont_manual <- cont_manual + 1
  }
  if (temp_nativa[i] > 68){
    cont_nativa <- cont_nativa + 1
  }
}

cat("P(T>68) manual empirica  : ", cont_manual/1000, "\n")
cat("P(T>68) nativa empirica  : ", cont_nativa/1000, "\n")

#TEORICA
prob_teo_68 <- 1 - pnorm(68, mean = 62, sd = 3.5)
cat("P(T>68) teorica  : ", prob_teo_68, "\n")

#(e)P(60 <T <65)
#EMPIRICA
cont_manual <- 0
cont_nativa <- 0

for(i in 1:1000){
```

```
  if(temp_manual[i] > 60 & temp_manual[i] < 65){
    cont_manual <- cont_manual + 1
  }
  if (temp_nativa[i] > 60 & temp_nativa[i] < 65){
    cont_nativa <- cont_nativa + 1
  }
}

cat("P(60 < T < 65) manual empirica  : ", cont_manual/1000, "\n")
cat("P(60 < T < 65) nativa empirica  : ", cont_nativa/1000, "\n")

#TEORICA
prob_teo_6065 <- pnorm(65, mean = 62, sd = 3.5) - pnorm(60, mean = 62, sd
    = 3.5)
cat("P(60 < T < 65) teorica  : ", prob_teo_6065, "\n")
#item 4
#(a)


hist(temp_manual, freq = FALSE, main = "Temperatura manual e nativa
    sobrepostas", xlab = "Valor", ylab = "Densidade", col = "lightblue",
    border = "white")
hist(temp_nativa, freq = FALSE, add = TRUE, col = rgb(1, 0, 0, 0.5),
    border = "white") # rgb(R, G, B, alpha)

curve(dnorm(x, mean = 62, sd = 3.5),
      add = TRUE,
      col = "blue",
      lwd = 2,
      lty = 2)

#(f)P(T>75)
#TEORICO
prob_teo_75 <- 1 - pnorm(75, mean = 62, sd = 3.5)
cat("P(T>75) teorica  : ", prob_teo_75, "\n")
```