

Código R para o exercício 4

```
# Parametros da distribuicao Weibull
lambda <- 29
k <- 7

# 1. Calculo exato de E(X)
E_exato <- lambda * gamma(1 + 1 / k)

# 2. Gerar a amostra e o valor empirico de E(X)
set.seed(1160)
n <- 5000
amostra <- rweibull(n, shape = k, scale = lambda)
E_empirico <- mean(amostra)

# 3. Desvio absoluto arredondado
desvio <- abs(E_exato - E_empirico)
desvio_arredondado <- round(desvio, 4)

# Print do resultado
#cat("Diferenca absoluta:", desvio_arredondado, "\n")
cat("Resultado do exercicio 4 = ", desvio_arredondado, "\n")
```

Código R para o exercício 5

```
# Semente dada
set.seed(1091)

# Numero de jogadas a simular (row)
nr_lan <- 30000
# Numero de dados por jogada (col)
nr_dice <- 3

# Simular todos os lancamentos: matriz n x nr_dice
# Usando o matrix gera todos os resultados de uma vez, o que seria mais
# eficiente do que correr uma funcao varias vezes
dice_rolls <- matrix(sample(1:6, nr_lan * nr_dice, replace = TRUE), nrow =
  nr_lan, ncol = nr_dice)

# Soma dos dados por jogada
somas <- rowSums(dice_rolls)

# Frequencias relativas das somas 9 e 10
freq_9 <- mean(somas == 9)
freq_10 <- mean(somas == 10)

# Diferenca com 4 casas decimais
diferenca <- round(abs(freq_10 - freq_9), 4)
# Print dos resultados
#print(diferenca)
cat("Resultado do exercicio 5 = ", diferenca, "\n")
```

Código R para o exercício 6

```
#                                     PASSO 1: VALOR EXATO DE  $P_n$ 
x <- 5.3
n <- 11
valor_exato_pn <- function(n, x) {
  floor_x <- floor(x)
  soma_termos <- 0
  for (k in 0:floor_x) {
    termo <- (-1)^k * choose(n, k) * (x - k)^n
    soma_termos <- soma_termos + termo
  }
  return(soma_termos / factorial(n))
}
pn_exact <- valor_exato_pn(n, x)
#                                     PASSO 2: SIMULACAO COM A SEED=3985
#                                     E
#                                     APROXIMACAO PELO TLC (TEOREMA DO LIMITE CENTRAL)
#=====
# COM A SEED
#=====
set.seed(3985)
m <- 110
sn_values <- numeric(m)
for (i in 1:m) {
  sample_x <- runif(n, 0, 1)
  sn_values[i] <- sum(sample_x)
}
pn_sim <- mean(sn_values <= x)
#=====
# COM TLC
#=====
mu_sn <- n * 0.5
var_sn <- n * (1/12)
sd_sn <- sqrt(var_sn)
z <- (x - mu_sn) / sd_sn
pn_tlc <- pnorm(z)
#                                     PASSO 3: DESVIO ABSOLUTO ENTRE  $P_n$  E  $P_{n,tlc}$ 
desvio_tlc <- abs(pn_exact - pn_tlc)
#                                     PASSO 4: DESVIO ABSOLUTO ENTRE  $P_n$  e  $P_{n,sim}$ 
desvio_sim <- abs(pn_exact - pn_sim)
#                                     PASSO 5: QUOCIENTE ENTRE OS DESVIOS
quociente <- desvio_tlc / desvio_sim
#                                     MOSTRAR RESULTADOS COM 4 CASAS DECIMAIS
#cat("1. Valor exato  $p_n$  =", round(pn_exact, 4), "\n")
#cat("2a. Aproximacao TLC  $p_{n-TLC}$  =", round(pn_tlc, 4), "\n")
#cat("2b. Aproximacao por simulacao  $p_{n-sim}$  =", round(pn_sim, 4), "\n")
#cat("3. Desvio TLC = ", round(desvio_tlc, 4), "\n")
#cat("3. Desvio da simulacao = ", round(desvio_sim, 4), "\n")
#cat("4. Quociente = ", round(quociente, 4), "\n")
cat("Resultado do exercicio 6 = ", round(quociente, 4), "\n")
```

Código R para o exercício 7

```
# Dados do enunciado
n <- 12
soma_x <- 61.17
soma_logx <- 19.45

# Medias
media_x <- soma_x / n
media_logx <- soma_logx / n

# Funcao para encontrar a raiz (equacao de verossimilhanca para alpha)
veross <- function(alpha) {
  log(alpha) - digamma(alpha) - log(media_x) + media_logx
}
resultado <- uniroot(veross, interval = c(46.6, 63.4))
alpha_hat <- resultado$root

# Estimar lambda
lambda_hat <- alpha_hat / media_x
modo <- (alpha_hat - 1) / lambda_hat

# Mostrar resultado com 2 casas decimais
#round(modo, 2)
cat("Resultado do exercicio 7 = ", round(modo, 2), "\n")
```

Código R para o exercício 8

```
# Dados do enunciado
set.seed(1308)
m <- 1300
n <- 10
mu <- 0.5
sigma <- 1
gamma <- 0.91
z <- qnorm(1 - (1 - gamma) / 2)

# Simulacao
cobertura <- replicate(m, {
  amostra <- rnorm(n, mean = mu, sd = sigma)
  media <- mean(amostra)
  erro <- z * sigma / sqrt(n)
  li <- media - erro
  ls <- media + erro
  mu >= li && mu <= ls
})

# Proporcao de intervalos que contem o valor verdadeiro
proporcao <- mean(cobertura)

# Quociente
quociente <- proporcao / gamma

# Resultado final com 4 casas decimais
#round(quociente, 4)
cat("Resultado do exercicio 8 = ", round(quociente, 4), "\n")
```

Código R para o exercício 9

```
# Dados do enunciado
set.seed(5707)

m <- 900          #numero de amostras
n <- 25           #dimensao da populacao X
alfa <- 0.08
h0 <- 5
h1 <- 6.2
# Aqui calculamos o valor critico
# Este e o limiar de decisao (rejeitamos h0 de T0>critico)
c <- qchisq(1 - alfa, df = 2 * n)
# Inicializar o contar de erros_tipo_II
# E incrementado quando nao rejeitamos h0 (mesmo que he=True)
contador_erros_II <- 0
#Loop que corre n vezes
for (i in 1:m) {
  #gera uma amostra de tamanho n=25 com media h1=6.2
  amostra <- rexp(n, rate = 1 / h1) # gerar sob H1
  #calculamos a media amostral usada na estatistica
  media_amostral_X <- mean(amostra)
  #calculamos a estatistica T0
  T0 <- (2 * n * media_amostral_X) / h0
  #aqui verificamos se nao ultrapassa o critico
  #incrementamos o numero de erros_tipo_II caso nao ultrapasse
  if (T0 <= c) {
    contador_erros_II <- contador_erros_II + 1
  }
}
#dividir o numero de erros_tipo_II pelo numero total de simulacoes
beta_chapeu <- contador_erros_II / m
# Calcular valor teorico de beta
beta <- pgamma(c, shape = n, scale = 2 * h1/h0)
# Quociente
quociente <- round(beta_chapeu/beta, 4)
#print(quociente)
cat("Resultado do exercicio 9 = ", quociente, "\n")
```

Código R para o exercício 10

```
# 1. Dados da amostra
dados <- c(2.3, 2.7, 5.2, 0.7, 2.9, 0.6, 2.6, 2.2, 3.8, 0.5, 4.9, 5.4,
  3.7, 0.4, 4, 3.6, 2, 0.8, 2.5, 2.8, 1.7, 3.3,
  1.5, 0.4, 6.4, 1.5, 6, 2.1, 0.4, 4.6, 3.1, 4.4, 4, 2.1, 5, 3.3,
  4.7, 3.4, 4.3, 4.5, 2.3, 0.5, 4.9, 3.5,
  1.8, 1.9, 2.6, 4.3, 4.6, 5.2, 1.6, 2.8, 2.4, 2.8, 1.8, 3.6,
  0.8, 5.1, 1.4, 3.2, 1, 6.3, 3.6, 3.6, 1.8,
  0.9, 4.6, 2.5, 5.8, 0.6, 3.3, 3.2, 6.6, 2.6, 2.5, 1.5, 4.1,
  1.7, 2.1, 1.5, 0.4, 4.8, 0.4, 1.5, 4.2, 3.3,
  1.2, 8.1, 2.4, 2.8, 2.1, 6.3, 4.2, 1.3, 6, 1.3, 3.7, 2.5, 6.6,
  2.7, 1.4, 2, 0.7, 4.3, 3.4, 4.3, 4, 4,
  0.8, 2.3, 2.5, 5.4, 4.3, 0.5, 3.9, 2.2, 3.4, 1.3, 2.4, 4.7, 2,
  1.3, 4.4, 2.9, 2.1, 2.5, 1.6, 2.3, 4.4,
  1.9, 1.9, 1.7, 2, 4.2, 3.4, 3.9, 4.3, 1.3, 2.9, 2.2, 5.1, 2.3,
  1.9, 2.9, 5.2, 3.4, 2.6, 2.4, 3.2, 1.3,
  3.1, 5.1, 1.4, 4.2, 0.9, 1.3, 2.1, 2.6, 6.2, 1.6, 2.7, 1.7,
  2.3, 3.3, 2.8, 1.2, 2.6, 1.5, 2, 2.8, 2.5, 2,
  1.2, 2.2, 2.6, 2.5, 6, 1.9, 3, 3.8, 1.9, 3.2, 3.1, 1.8, 2.6,
  1.9, 3.5, 3.7, 1.8, 2.2, 2, 1.3, 2, 1.1,
  2.2, 3.1, 2.9, 1.3, 0.2, 3.9)

set.seed(4497)
n<-140
# Calcular limites das 6 classes equiprovaveis sob H0: Rayleigh(sigma=2.2)
sigma <- 2.2
k <- 6
indices_sub <- sample(1:length(dados), n, replace = FALSE)
subamostra <- dados[indices_sub]
probs <- seq(0.2, 1.0, by = 0.2)
limites <- sigma * sqrt(-2 * log(1 - probs))
for (i in 2:length(limites)) {
}
# Agrupar observacoes e contar frequencias observadas
freq_obs <- numeric(k)
for (x in subamostra) {
  if (x <= limites[1]) {
    freq_obs[1] <- freq_obs[1] + 1
  } else if (x <= limites[2]) {
    freq_obs[2] <- freq_obs[2] + 1
  } else if (x <= limites[3]) {
    freq_obs[3] <- freq_obs[3] + 1
  } else if (x <= limites[4]) {
    freq_obs[4] <- freq_obs[4] + 1
  } else {
    freq_obs[5] <- freq_obs[5] + 1
  }
}
# Frequencias esperadas
freq_esp <- n / k
# Teste qui-quadrado
chi2_stat <- sum((freq_obs - freq_esp)^2 / freq_esp)
gl <- k - 1
p_valor <- 1 - pchisq(chi2_stat, gl)
```

```
# Decisoes nos niveis de significancia
niveis <- c(0.01, 0.05, 0.10)
decisoes <- character(length(niveis))
for (i in 1:length(niveis)) {
  if (p_valor < niveis[i]) {
    decisoes[i] <- "Rejeitar H0"
  } else {
    decisoes[i] <- "Nao rejeitar H0"
  }
}
cat("\ nDecisoes :\n")
cat(" Nivel 1%: ", decisoes [1] , "\n")
cat(" Nivel 5%: ", decisoes [2] , "\n")
cat(" Nivel 10%: ", decisoes [3] , "\n")
# Determinar a resposta correta
if (all(decisoes == "Nao rejeitar H0")) {
  resposta <- "c"
  cat("\nResposta: c) Nao rejeitar H0 aos n.s. de 1%, 5% e 10%.\n")
} else if (all(decisoes == "Rejeitar H0")) {
  resposta <- "d"
  cat("\nResposta do exercicio 10: : d) Rejeitar H0 aos n.s. de 1%, 5% e
    10%.\n")
} else if (decisoes[1] == "Nao rejeitar H0" && decisoes[2] == "Rejeitar H0"
  && decisoes[3] == "Rejeitar H0") {
  resposta <- "a"
  cat("\nResposta do exercicio 10: : a) Rejeitar H0 aos n.s. de 5% e 10% e
    nao rejeitar H0 ao n.s. de 1%.\n")
} else if (decisoes[1] == "Nao rejeitar H0" && decisoes[2] == "Nao
  rejeitar H0" && decisoes[3] == "Rejeitar H0") {
  resposta <- "b"
  cat("\nResposta do exercicio 10: : b) Rejeitar H0 ao n.s. de 10% e nao
    rejeitar H0 aos n.s. de 1% e 5%.\n")
} else {
  resposta <- "e"
  cat("\nResposta do exercicio 10: : e) Teste e inconclusivo.\n")
}
```