Cupom Coletor e Paradoxo do Aniversário

Code

Hide

```
#Pacotes utilizados
library(ggplot2)
```

```
Aviso: pacote 'ggplot2' foi compilado no R versão 4.4.3
```

Hide

```
#seed fixa
set.seed(123)
```

Introdução

Neste projeto estudaremos dois problemas de probabilidade clássicos, o cupom-coletor e o paradoxo do aniversário. O objetivo do trabalho é utilizar técnicas de programação para simular em grandes iterações e diferentes métricas os problemas, e visa coletar resultados empíricos e comparar com os resultados teóricos.

Cupom-coletor

Neste problema, o colecionador de cupons tem como objetivo completar o seu "álbum". No contexto, o colecionador realiza sorteios com reposição dos itens do álbum até completar a sua coleção. A pegunta para este problema é "espera-se quantos sorteios com reposição para completar todos os itens da coleção?"

Simulação: Cupom-coletor

Considere um álbum com N itens distintos. Em cada passo, um itens será sorteado do conjunto, sendo que cada item tem probabilidade $1/N$ de ser escolhido. O item é reposto e o sorteio é feito novamente. O processo continua até que todos os N itens é observado pelo menos uma vez, ou seja, o "álbum é completado". A variável de interesse é o número de sorteios necessários T_n.

Hide

```
CupCol <- function(N){
  aux <- c() #vetor auxiliar para armazenar os itens já sorteados
  Tn <- 0 #para contar o número de sorteios realizados

  while(length(aux) < N){ #loop até completar o álbum
    Tn <- Tn + 1 #contagem dos sorteios
    i <- sample(1:N, 1) #sorteio de um item com probabilidades iguais

    if(i %in% aux){ #verifica se o item já foi observado
      next #pula para próxima iteração
    }
    else{ #se o item não foi observado ainda
      aux <- c(aux, i) # adiciona o item no vetor aux
    }
  }
  return(Tn) #retorna o numero de sorteios feitos
}
```

Acima, escrevemos uma função o qual simula o problema do cupom-coletor, com parâmetro N. Podemos simular o problema com diferentes Ns.

Hide

```
CupCol(N = 50)
```

```
[1] 129
```

Hide

```
CupCol(N = 100)
```

```
[1] 316
```

Hide

```
CupCol(N = 200)
```

```
[1] 2082
```

Hide

```
CupCol(N = 500)
```

```
[1] 3034
```

Vamos realizar $M = 5000$ réplicas para cada N.

Hide

```
replicador <- function(M, N){
  aux <- c()
  for(i in 1:M){ #faz M iteracoes do processo
    x <- CupCol(N) #chama a função
    aux <- c(aux, x) #armazena o valor x em um vetor
  }
  return(aux)#retorna o vetor com os M numeros de sorteios

}
```

Hide

```
#M = 5000 para todo N
A50 <- replicador(5000, 50) #N=50
A100 <- replicador(5000, 100) #N=100
A200 <- replicador(5000, 200) #N=200
A500 <- replicador(5000, 500) #N=500
```

Assim, podemos encontrar a média, a variância e distribuição empírica das simulações:

Hide

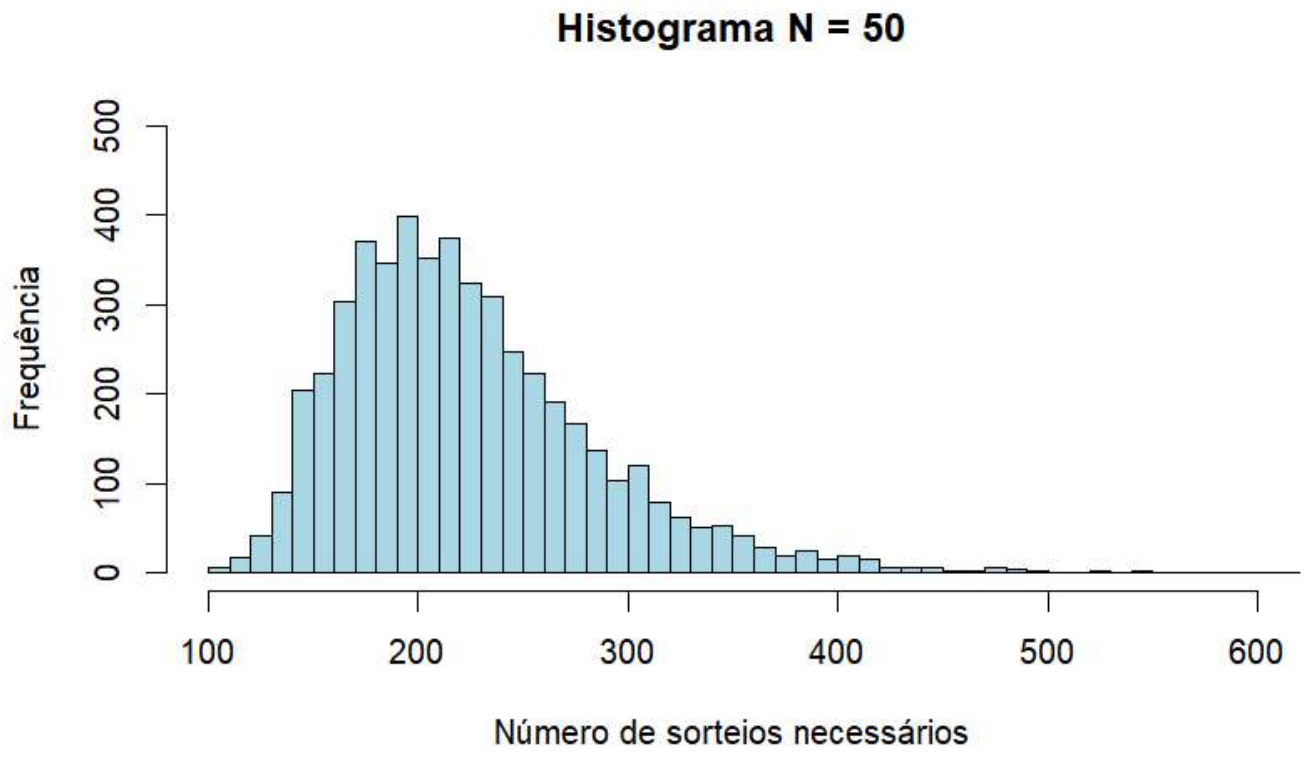
```
#Tabela de Médias e Variancias para cada N (empírico)
Num <- c("50", "100", "200", "500")
medias_emp <- c(mean(A50), mean(A100), mean(A200), mean(A500))
variancias_emp <- c(var(A50), var(A100), var(A200), var(A500))

tabela_emp <- data.frame(
  N = Num,
  Média = round(medias_emp, 2),
  Variância = round(variancias_emp, 2)
)

print(tabela_emp)
```

N <chr>	Média <dbl>	Variância <dbl>
50	225.14	3909.20
100	520.10	15751.10
200	1181.65	66575.11
500	3397.41	415122.04

4 rows

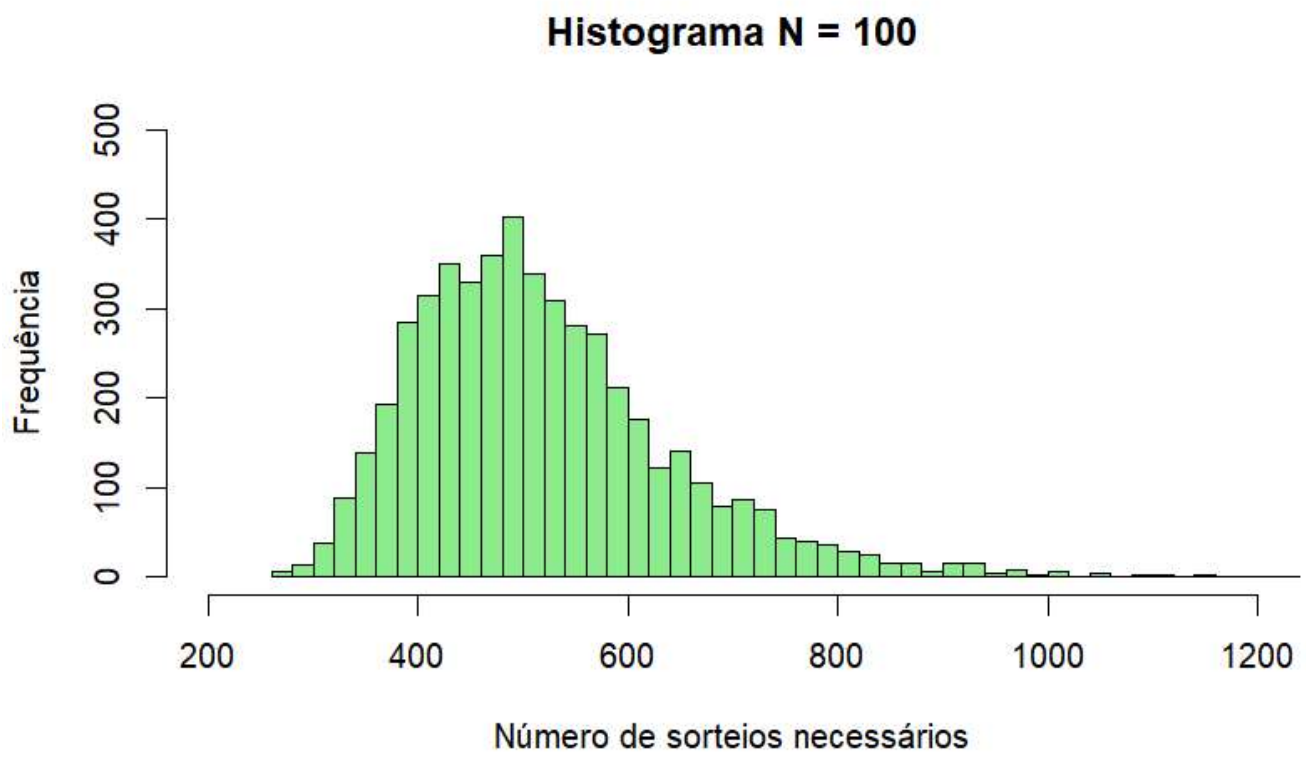
Hide

```
h_emp_50 <- hist(A50,
     main = "Histograma N = 50",
     xlab = "Número de sorteios necessários",
     ylab = "Frequência",
     col = "lightblue",
     border = "black",
     breaks = 50,
     xlim = c(100, 600),
     ylim = c(0, 500))
```



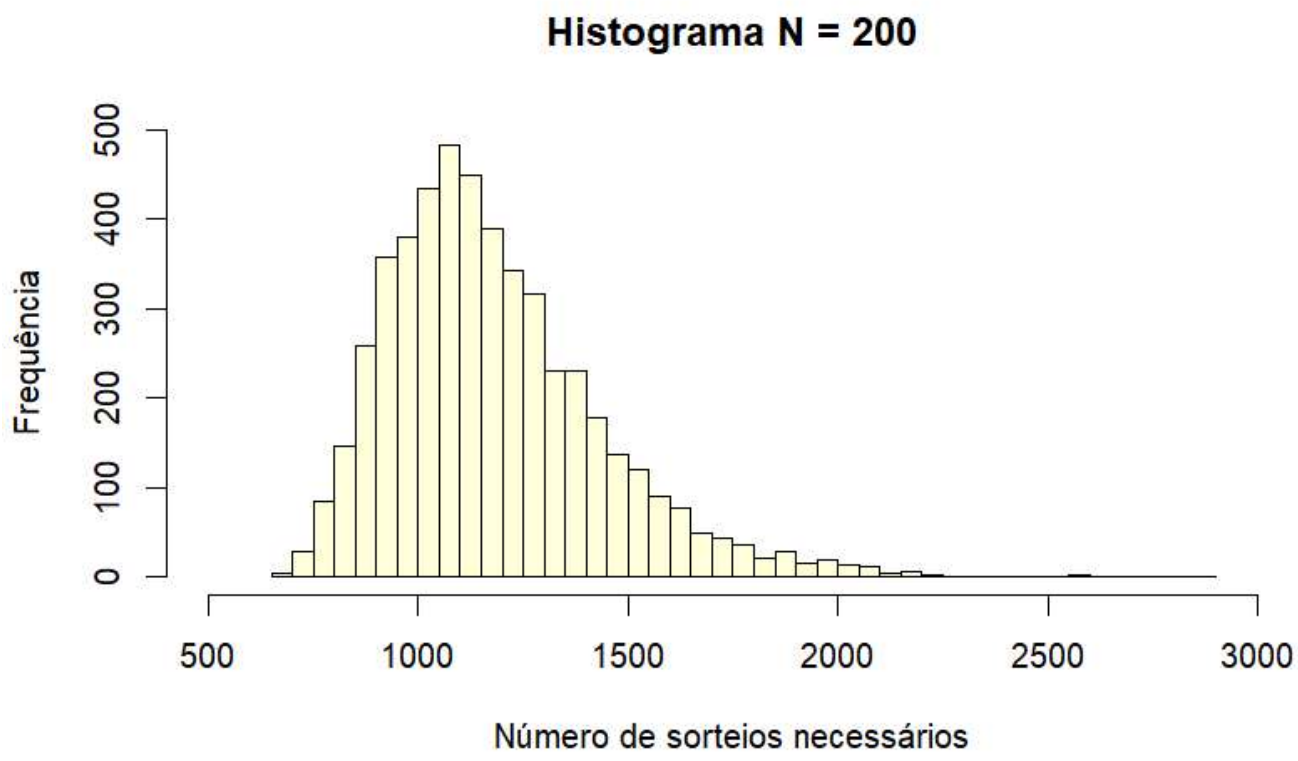
Hide

```
h_emp_100 <- hist(A100,
     main = "Histograma N = 100",
     xlab = "Número de sorteios necessários",
     ylab = "Frequência",
     col = "lightgreen",
     border = "black",
     breaks = 50,
     xlim = c(200, 1200),
     ylim = c(0, 500))
```



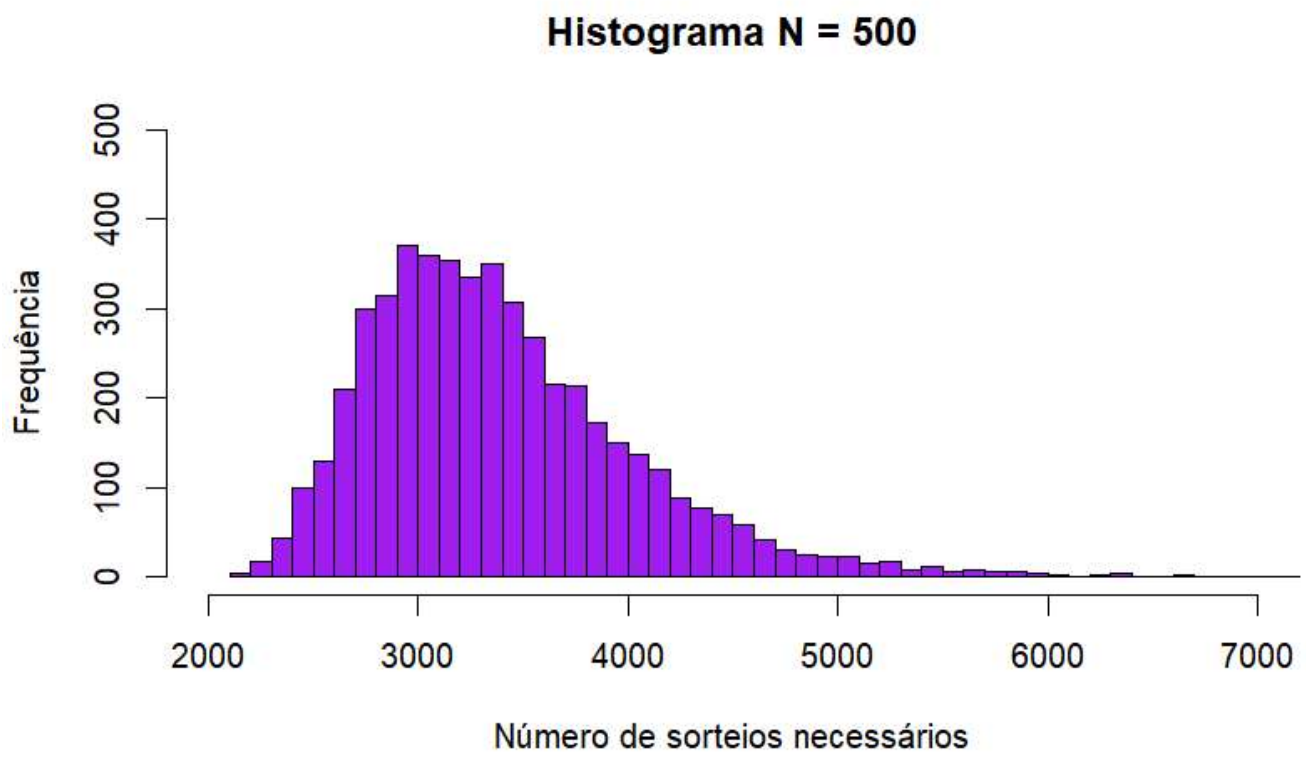
Hide

```
h_emp_200 <- hist(A200,
     main = "Histograma N = 200",
     xlab = "Número de sorteios necessários",
     ylab = "Frequência",
     col = "lightyellow",
     border = "black",
     breaks = 50,
     xlim = c(500, 3000),
     ylim = c(0, 500))
```



Hide

```
h_emp_500 <- hist(A500,
     main = "Histograma N = 500",
     xlab = "Número de sorteios necessários",
     ylab = "Frequência",
     col = "purple",
     border = "black",
     breaks = 50,
     xlim = c(2000, 7000),
     ylim = c(0, 500))
```



Hide

```
NA
NA
```

Teoria: Cupom-coletor

O valor esperado do T (número de sorteios) é (resultados em(1)):

$$E(T) = nH(n)$$

onde o $H(n)$ é o n-ésimo número harmônico:

$$H(n) = 1 + \frac{1}{2} + \ldots + \frac{1}{n-1} + \frac{1}{n}$$

Para valores grandes de n, pode ser aproximado como:

$$E(T) = nH(n) \approx n\ln(n) + \gamma n + \frac{1}{2} + O\left(\frac{1}{n}\right)$$

onde $\gamma \approx 0.577215665$ é a constante Euler-Mascheroni e $O\left(\frac{1}{n}\right)$ representa um termo de erro que decresce proporcionalmente a $1/n$.

A variância de T é (resultados em (1)):

$$Var(T) < n^2\left(\frac{\pi^2}{6}\right)$$

Assim, podemos encontrar o valor esperado e variância aproximados para cada n:

Hide

```
#funçoes para calcular a medias e variancias teoricas
media <- function(n){
  return(n*log(n) + 0.577215665*n + 1/2 )
}
variancia <- function(n){
  return((n^2)*(pi^2)/6)
}
```

Hide

```
#Tabela de medias e variancias aprox. (Teorica)
medias_real <- c(media(50), media(100), media(200), media(500))
variancias_real <- c(variancia(50), variancia(100), variancia(200), variancia(500))

tabela_real <- data.frame(
  N = Num,
  Média = round(medias_real, 2),
  Variância = round(variancias_real, 2)
)

print(tabela_real)
```

N <chr>	Média <dbl>	Variância <dbl>
50	224.96	4112.34
100	518.74	16449.34
200	1175.61	65797.36
500	3396.41	411233.52

4 rows

Para mais detalhes, fazemos o teste-t de student para a média empírca com a média real:

Hide

```
t.test(A50, mu = media(50))
```

```

    One Sample t-test

data:  A50
t = 0.19822, df = 4999, p-value = 0.8429
alternative hypothesis: true mean is not equal to 224.9619
95 percent confidence interval:
 223.4037 226.8707
sample estimates:
mean of x
 225.1372
```

Hide

```
t.test(A100, mu = media(100))
```

```
    One Sample t-test

data:  A100
t = 0.76535, df = 4999, p-value = 0.4441
alternative hypothesis: true mean is not equal to 518.7386
95 percent confidence interval:
 516.6174 523.5766
sample estimates:
mean of x
  520.097
```

Hide

```
t.test(A200, mu = media(200))
```

```
    One Sample t-test

data:  A200
t = 1.656, df = 4999, p-value = 0.09779
alternative hypothesis: true mean is not equal to 1175.607
95 percent confidence interval:
 1174.496 1188.803
sample estimates:
mean of x
 1181.649
```

Hide

```
t.test(A500, mu = media(500))
```

```
    One Sample t-test

data:  A500
t = 0.10919, df = 4999, p-value = 0.9131
alternative hypothesis: true mean is not equal to 3396.412
95 percent confidence interval:
 3379.544 3415.270
sample estimates:
mean of x
 3397.407
```

Empírico x Teórico

Podemos comparar os resultados obtidos empiricamente e os teóricos:

Hide

```
#tabela com empirico e teorico
tabela_final <- data.frame(
  N = Num,
  Média_emp = round(medias_emp, 2),
  Média_teo = round(medias_real, 2),
  Variância_emp = round(variancias_emp, 2),
  Variância_teo = round(variancias_real, 2)

)
print(tabela_final)
```

N <chr>	Média_emp <dbl>	Média_teo <dbl>	Variância_emp <dbl>	Variância_teo <dbl>
50	225.14	224.96	3909.20	4112.34
100	520.10	518.74	15751.10	16449.34
200	1181.65	1175.61	66575.11	65797.36
500	3397.41	3396.41	415122.04	411233.52

4 rows

Encontramos também o erro absoluto:

Hide

```
erro_media <- abs(medias_emp - medias_real)
erro_variancia <- abs(variancias_emp - variancias_real)

tabela_erro <- data.frame(
  N = Num,
  Médias_erro = erro_media,
  Variâncias_erro = erro_variancia
)
print(tabela_erro)
```

N <chr>	Médias_erro <dbl>	Variâncias_erro <dbl>
50	0.1752665	203.1346
100	1.3584149	698.2365
200	6.0425937	777.7489
500	0.9949183	3888.5250

4 rows

Paradóxo do aniversário

Neste problema, em grupo de k indivíduos aleatórios, queremos encontrar a probabilidade de duas pessoas ter a mesma data de aniversário.

Simulação: Paradóxo do aniversário

Fixe $N = 365$. Para cada $k \in [2, 60]$ pessoas será feito um sorteio com reposição de 1 até N, k vezes. Se tiver um item já observado, o sorteio conta como sucesso. Se não, fracasso. A variável de interesse é um indicador booleano se foi sucesso ou fracasso no sorteio.

Hide

```
ParAni <- function(N, k){
  aux <- c()#vetor auxiliar para armazenar as datas sorteadas
  bool <- FALSE# booleano para ver se foi sucesso ou fracasso
  for(i in 1:k){ #loop para sortear as datas k vezes
    x <- sample(1:N, 1) #sorteio de uma data, de 1 até N=365
    if(x %in% aux){ #verifica se houve datas repetidas(colisão)
      bool <- TRUE #booleano torna true=sucesso
      break #termina as iteraçoes
    }
    else{
      aux <- c(aux, x)#se não, adiciona a data no vetor aux
    }

  }
  return(bool)#retorna o booleano
}
```

Faremos a simulação com $N = 365$ fixo, para diferentes k :

Hide

```
N <- 365
ParAni(N, 2)#k=2
```

```
[1] FALSE
```

Hide

```
ParAni(N, 20)#k=20
```

```
[1] TRUE
```

Hide

```
ParAni(N, 350)#k=350
```

```
[1] TRUE
```

Para cada $k \in [2 : 60]$, realizaremos $M = 10000$ simulações:

Hide

```
repetidor <- function(k, M){
 aux <- list() #lista para armazenar os vetores
for(i in 2:k){ #para cada k, sera feito os loops
  vetor <- c() #vetor para armazenar os booleanos
  for(l in 1:M){
   x <- ParAni(N = 365, i) #sorteios
   vetor <- c(vetor, x) #adiciona o novo bool no vetor
  }
  aux <- c(aux, list(vetor)) #adiciona o vetor na lsta
  }
return(aux)#retorna a lista
 }
```

Hide

```
k <- 60
M <- 10000
PA_60 <- repetidor(k, M)
```

Vamos encontrar a frequência de sucesso(colisão) para cada k :

Hide

```
f <- function(lista, M){
n <- length(lista)
contagem_true <- c()#vetor para armazenar quantos true tem por k
for(i in 1:n){
  true_count <- sum(unlist(lista[[i]]) %in% TRUE, na.rm = TRUE)
  contagem_true <- c(contagem_true, true_count)
  }
porc_true <- (contagem_true/M)*100 #porcentagem da taxa de sucesso
return(porc_true)
}

x <- f(PA_60, M)
```

Para visualizar melhor, faremos uma tabela e um gráfico com os dados obtidos experimentalmente:

Hide

```
tabela_freq <- data.frame(
  N = 2:k,
  f_sucesso = x
)
print(tabela_freq)
```

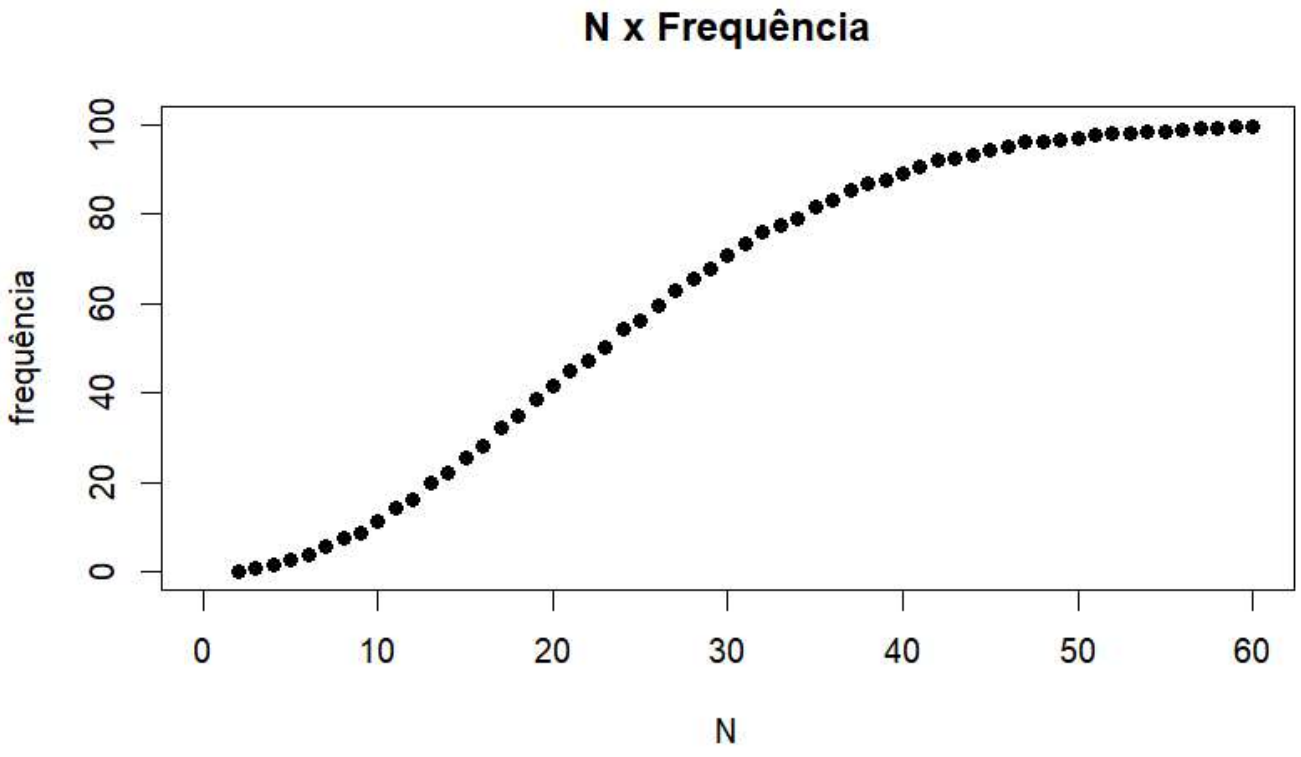
N <int>	f_sucesso <dbl>
2	0.25
3	0.76
4	1.60
5	2.78

N <int>	f_sucesso <dbl>
6	3.95
7	5.92
8	7.55
9	8.81
10	11.28
11	14.41

1-10 of 59 rows Previous **1** 2 3 4 5 6 Next

Hide

```
t_ani_emp <- plot(tabela_freq$N, tabela_freq$f_sucesso,
     main = "N x Frequência",
     xlab = "N",
     ylab = "frequência",
     col = "black",
     pch = 16,
     cex = 1,
     type = "p",
     xlim = c(0, 60),
     ylim = c(0, 100)
)
```



Teoria: Paradoxo do Aniversário

Para calcular a probabilidade de n pessoas em uma sala , em que pelo menos duas pessoas tenham o mesmo aniverário, calculamos a probabilidade do complementar, ou seja, a probabilidade de que n pessoas não tenham o mesmo aniversário(referencias em 2):

$$\bar{p}(n) = 1 \cdot \left(1 - \frac{1}{365}\right) \cdot \left(1 - \frac{2}{365}\right) \cdots \left(1 - \frac{n-1}{365}\right) = \frac{365!}{365^n (365-n)!}$$

Com isso, encontramos a probabilidade desejada, de n pessoas, pelo menos duas pessoas terem o mesmo aniversário:

$$p(n) = 1 - \bar{p}(n)$$

Fazemos então o código para encontrar as probabilidades:

Hide

```
f_prob_an <- function(n){
  if(n > 365){
    return(1) #se n maior que 365, a probabilidade é 1
  }
  else if (n < 2){
    return(0) #se n menor que 2,  a probabilidade é 0
  }
  else {
    prob_diferentes <- 1
    for (i in 1:(n-1)) {
      prob_diferentes <- prob_diferentes * (365 - i) / 365
    }
    # Probabilidade de pelo menos 2 terem o mesmo aniversário
    return(1 - prob_diferentes)
  }
}
```

Hide

```
k = 60
vetor_aniv <- c()
for(i in 2:k){
  y <- f_prob_an(i)
  vetor_aniv <- c(vetor_aniv, y)
}

tabela_freq_real <- data.frame(
  N = 2:k,
  f_sucesso_real = vetor_aniv *100
)
print(tabela_freq_real)
```

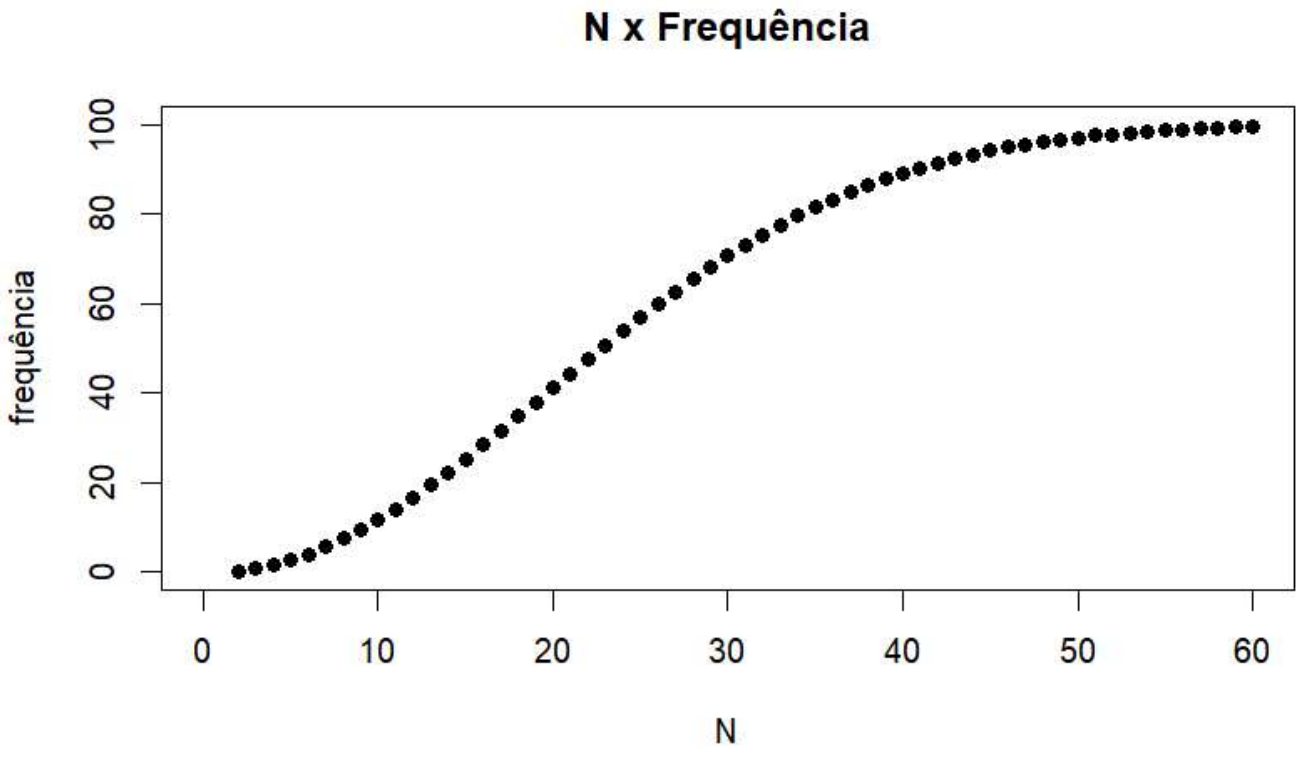
N <int>	f_sucesso_real <dbl>
2	0.2739726
3	0.8204166
4	1.6355912
5	2.7135574

N <int>	f_sucesso_real <dbl>
6	4.0462484
7	5.6235703
8	7.4335292
9	9.4623834
10	11.6948178
11	14.1141378

1-10 of 59 rows Previous **1** 2 3 4 5 6 Next

Hide

```
t_ani_real <- plot(tabela_freq_real$N, tabela_freq_real$f_sucesso_real,
     main = "N x Frequência",
     xlab = "N",
     ylab = "frequência",
     col = "black",
     pch = 16,
     cex = 1,
     type = "p",
     xlim = c(0, 60),
     ylim = c(0, 100)
)
```



Empírico x teórico

Podemos comparar os resultados obtidos empiricamente e os teóricos, com tabela e gráfico:

Hide

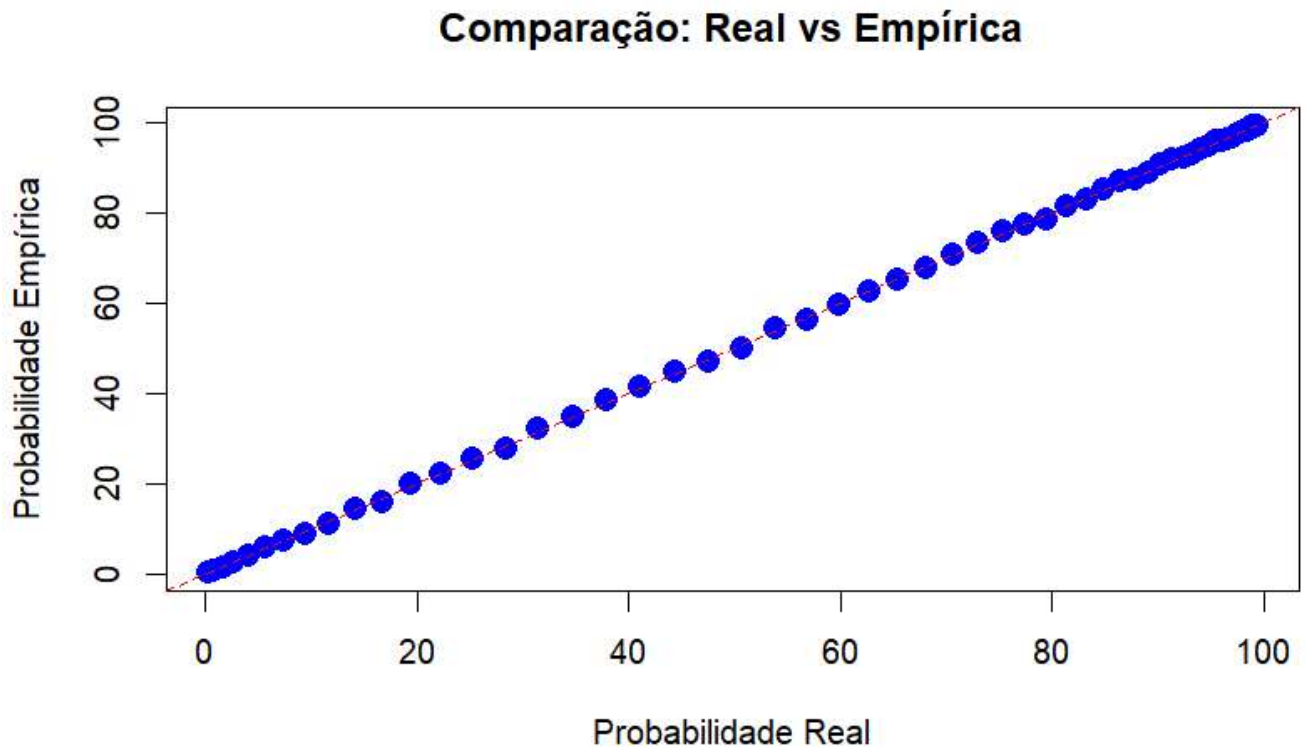
```
tabela_freq_geral <- data.frame(
  N = 2:k,
  f_sucesso_real = vetor_aniv *100,
  f_suceeso_emp = x
)
print(tabela_freq_geral)
```

N <int>	f_sucesso_real <dbl>	f_suceeso_emp <dbl>
2	0.2739726	0.25
3	0.8204166	0.76
4	1.6355912	1.60
5	2.7135574	2.78
6	4.0462484	3.95
7	5.6235703	5.92
8	7.4335292	7.55
9	9.4623834	8.81
10	11.6948178	11.28
11	14.1141378	14.41

1-10 of 59 rows Previous **1** 2 3 4 5 6 Next

Hide

```
plot(tabela_freq_real$f_sucesso_real, tabela_freq$f_sucesso,
     xlab = "Probabilidade Real",
     ylab = "Probabilidade Empírica",
     main = "Comparação: Real vs Empírica",
     pch = 16, col = "blue", cex = 1.5)
abline(a = 0, b = 1, col = "red", lty = 2)
```



Conclusões

Percebemos no projeto que, simulando um problema em algoritmo e obter uma grande quantidade de amostras independentes, podemos encontrar médias, variâncias ou probabilidades empíricas que tem boa precisão. Esse método é chamado de Método do Monte Carlo, que explica que ao obter amostras aleatórias de grande número, este tende ao parâmetro desejado, ou seja:

$$\hat{\theta_n} = \frac{1}{n} \sum g(X_i) \rightarrow \theta$$

quase certamente, pela Lei dos Grandes números.

Assim, concluímos que é possível obter respostas de problemas probabilísticos analicamente, ao simular o problema em códigos e obter várias amostras aleatórias, conforme foi feito no projeto.

Referências

1. https://towardsdatascience.com/coupon-collectors-problem-a-probability-masterpiece-1d5aed4af439/ (https://towardsdatascience.com/coupon-collectors-problem-a-probability-masterpiece-1d5aed4af439/)
2. https://pt.wikipedia.org/wiki/Paradoxo_do_anivers%C3%A1rio (https://pt.wikipedia.org/wiki/Paradoxo_do_anivers%C3%A1rio)