

```
# - Breve introdução ao R, operações com vetores e matrizes
#
# - Mostrar como simular variáveis aleatórias discretas e contínuas.
#
# - Avaliar estimadores
#
# - Gráficos, histogramas, boxplots, interpretação

# carregar bibliotecas -----

library(dplyr) # biblioteca para manipulacao e processamento de dados
library(ggplot2) # biblioteca para construcao de graficos

# lancamento de uma moeda honesta -----
rbinom(n = 1, size = 1, prob = 0.5) # funcao que amostra da distribuicao binomial

# lancamento de 10 moedas honestas -----
rbinom(n = 10, size = 1, prob = 0.5)

# semente aleatoria -----
set.seed(123)

rbinom(n = 10, size = 1, prob = 0.5)

# armazenar o resultado em uma variavel -----
set.seed(123)
lancamentos <- rbinom(n = 10, size = 1, prob = 0.5)

lancamentos # resultado

table(lancamentos) # tabela que conta os diferentes resultados

ifelse(lancamentos == 0, "cara", "coroa") # tranforma em caracteres

table(ifelse(lancamentos == 0, "cara", "coroa")) # obtem tabela

# estimar a proporcao de coroas -----
# calcular proporcao de coroas:
sum(lancamentos) / 10

# de forma mais elegante
sum(lancamentos) / length(lancamentos)

# ainda mais elegante
mean(lancamentos)

# construindo um grafico com o resultado -----

n <- 10 # numero total de lancamentos
p <- 0.5 # probabilidade de sucesso

set.seed(1234)
lancamentos <- rbinom(n = n, size = 1, prob = p)

dados <- data.frame(lancamento = 1:n,
                    resultado = lancamentos,
                    media = cumsum(lancamentos) / 1:n)

# construcao do grafico
ggplot(dados, aes(x = lancamento, y = media)) + # define as variaveis do eixo
  geom_line() + # adiciona uma geometria de linha
  geom_point() + # adiciona uma geometria de pontos
  geom_hline(yintercept = p, linetype = "dashed", color = "red") # adiciona uma linha horizontal

# -----
# TAREFA: aumente o valor de n (repeticoes) e veja o que acontece!
# -----
```

```

# Repetindo o procedimento para calcular o vies -----

set.seed(1) # define a semente aleatoria

N <- 10^(1:5) # vetor com o número de repeticoes do experimento
n <- 100 # número de lancamentos em cada repeticao
p <- 0.5 # probabilidade de sucesso

# vetores para armazenar o calculo dos estimadores do vies
vies_bar <- numeric(length(N))

for(j in 1:length(N)){
  x_bar <- numeric(N[j])
  for(i in 1:N[j]){
    lancamentos <- rbinom(n = n, size = 1, prob = p)
    x_bar[i] <- mean(lancamentos)
  }
  vies_bar[j] <- mean(x_bar) - p
}

# armazena os resultados em um dataframe
resultado <- data.frame(N = N, vies = vies_bar, estimador = "x_bar")

# visualiza o resultado graficamente
ggplot(resultado, aes(x = N, y = vies))+
  geom_line()+
  geom_point()

# -----
# TAREFA: construir um gráfico como o acima considerando um estimador para a variância.
# -----

# Variavel aleatoria normal -----
# vamos simular valores de salários em duas regioes diferentes, regio 1 e regio 2
set.seed(12)

n <- 1000 # numero de repeticoes

# gerar valores para regio 1
regiao1 <- data.frame(salario = rnorm(n, 2400, 100), regio = "Regiao 1")

# gerar valores para regio 2
regiao2 <- data.frame(salario = rnorm(n, 2600, 250), regio = "Regiao 2")

dados <- rbind(regiao1, regiao2) # empilha os dados em um unico dataframe

summary(regiao1) # sumario da regio 1
summary(regiao2) # sumario da regio 2

# histograma do salario
ggplot(dados, aes(x = salario)) +
  geom_histogram() +
  labs(x = "Salário", y = "Contagem")

# histograma do salario de cada regio
ggplot(dados, aes(x = salario, fill = regio)) +
  geom_histogram() +
  labs(x = "Salário", y = "Contagem", fill = "Região")

# histograma do salario de cada regio em graficos separados
ggplot(dados, aes(x = salario, fill = regio)) +
  geom_histogram() +
  facet_wrap(~regio)+
  labs(x = "Salário", y = "Contagem", fill = "Região")

# boxplot do salario de cada regio
ggplot(dados, aes(x = regio, y = salario, fill = regio)) +

```

```
geom_boxplot() +  
labs(x = "", y = "Salário")
```