

Introdução à Estatística – 1º Semestre 2017

Prof. Marcelo de Souza Lauretto



**EACH** |



Escola de Artes, Ciências e Humanidades  
Universidade de São Paulo

# Exercício-Programa 1

Por:

Leonardo Colman Lopes – nºUSP 9875490

Lucas Lacerda Pereira – nºUSP 9779047

O exercício programa 1 é efetuado através do código em R disponível em partes neste texto, o qual está devidamente documentado no email enviado em anexo para correção.

Inicialmente, é definido o  $n$  do programa. O  $n$  representa o número de portas disponíveis para o competidor escolher. O gráfico plotado em **Figura 1** representa um  $n = 3$  enquanto o gráfico plotado em **Figura 2** representa um  $n = 5$ .

Após definido o  $n$ , o programa prepara um vetor com 10.000 espaços para receber o resultado de 10.000 tentativas de abertura de portas, e executa todas as tentativas.

Cada tentativa consiste em:

- Popular um vetor de portas
- Escolher uma porta inicial
- Aguardar que o apresentador elimine portas
- Trocar a porta inicial pela porta que sobrou
- Colocar o conteúdo da porta escolhida por último no vetor de tentativas

### Popular vetor de portas:

Para popular o vetor de portas, o seguinte algoritmo é executado:

- Aloca-se um vetor do tamanho  $n$  pré-estabelecido
- Sorteia-se uma das portas aleatoriamente através da função **sample** para ser a porta que guarda a premiação
- Marca a porta sorteada como **1 (possui prêmio)** e as portas não sorteadas com **0 (não possui prêmio)**

### Escolher uma porta inicial

Para escolher uma porta inicial, ou seja, a porta que o participante escolheu enquanto todas as portas estavam fechadas, o seguinte algoritmo é executado:

- Sortear através da função **sample** um número aleatório, sendo este número algum número no intervalo  $[1, n]$

### Aguardar que o apresentador elimine portas

Tendo todas as portas preparadas, e uma escolhida aleatoriamente, o programa então executa uma função para que o “apresentador” abra e elimine  $n - 2$  portas. Este algoritmo é feito da seguinte forma:

- Cria uma variável para se medir a quantidade de portas abertas (iniciando em 0 portas abertas)
- Executa um loop de 1 até  $n$ , medido na variável  $p$ . Se o número de portas abertas já for  $n - 2$ , o loop para sua execução, caso contrário verifica-se se a porta que será aberta é a que contém o prêmio. Se for, ela não será aberta. Se a porta a ser aberta for a porta escolhida, ela também não será aberta. Caso contrário, a porta é aberta, marcando-a com 2.

## Trocar a porta inicial pela porta que sobrou aberta

Quando as portas estiverem abertas, o participante poderá escolher se quer trocar sua porta pela que permaneceu aberta, ou se prefere ficar com a própria porta escolhida inicialmente. A estratégia utilizada é de que o participante sempre trocará de porta. Portanto, é executada uma função para trocar a porta, que funciona da seguinte forma:

- Olha para todas as portas.
- Se a porta não estiver aberta, e também não for a própria porta, trocar
- Colocar na variável **segundaPortaEscolhida** o valor que está nessa porta, sendo **1 (possui prêmio)** e **0 (não possui prêmio)** os valores possíveis.
- Esse valor é colocado no vetor de tentativas

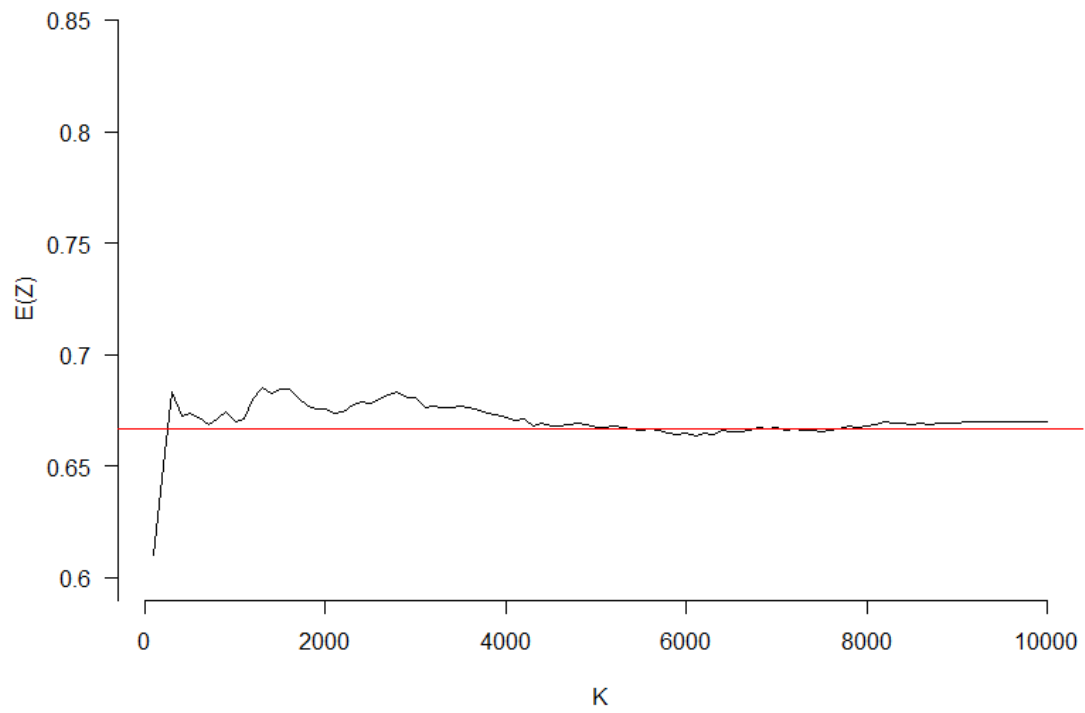
Quando as 10.000 tentativas forem finalizadas, o programa então prepara e imprime os dados para que se tornem legíveis ao ser humano. Para fazer isso, ele executa os seguintes passos:

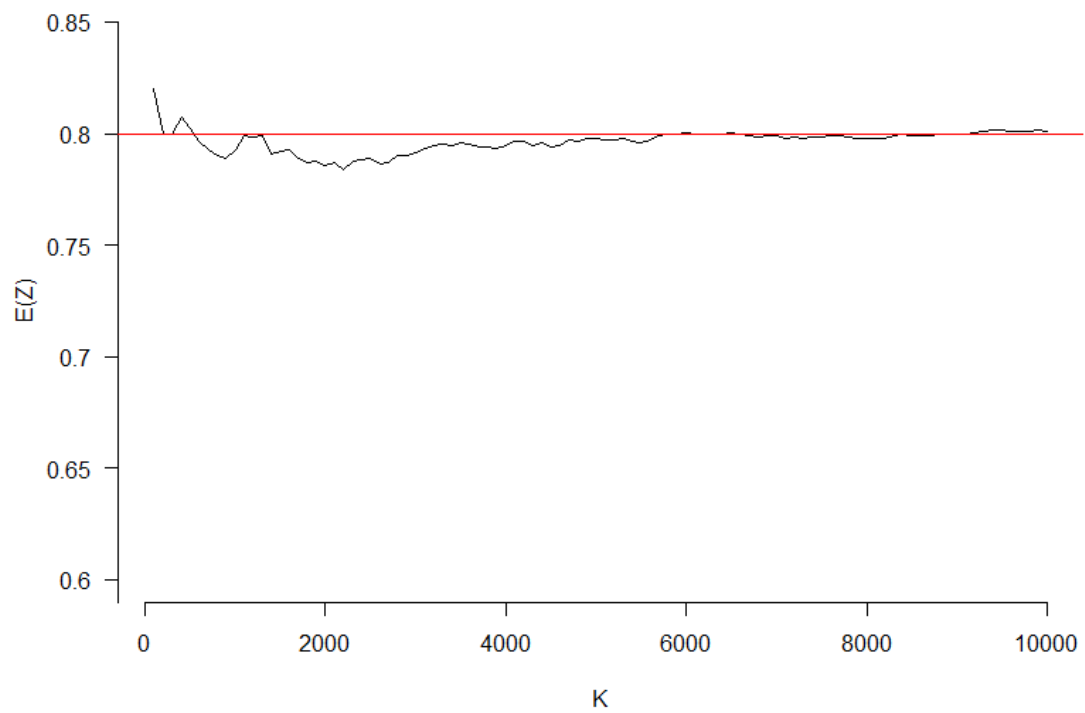
- Prepara um vetor de tamanho 100 para receber todas as médias parciais, que são calculadas da seguinte forma:
  - Cada posição no vetor recebe a média dos **100 \* index primeiros elementos** (index = posição do número no vetor), essencialmente, dos **k** primeiros elementos
- Faz o cálculo das médias parciais e imprime o último valor da média parcial, que representa a média total
- Desenha um gráfico do vetor com todas as médias parciais, e coloca uma linha do **valor verdadeiro**  $P r(Z) = (n - 1) / n$ . O vetor representado no gráfico é a convergência do número de tentativas ao valor verdadeiro.

Os gráficos que o programa efetua podem ficar diferentes, visto que as tentativas funcionam de maneira aleatória, mas dois exemplos podem ser conferidos no final deste arquivo. Pode-se verificar que ambos se aproximam mais do valor verdadeiro conforme as tentativas são feitas, para um número suficientemente grande de tentativas, os valores ficam bem próximos à linha.

Além disso, é visível que os valores começam a entrar nas proximidades do valor-verdadeiro a medida que o número de tentativas se aproxima de 4000.

**Figura 1** Convergência de  $\hat{\mu}_Z(k)$  para o valor verdadeiro  $P r(Z)=(n-1)/n$ , para  $n = 3$ . Média final:  
**0.6698**





**Figura 2** Convergência de  $\hat{\mu}_Z(k)$  para o valor verdadeiro  $P r(Z)=(n-1)/n$ , para  $n = 5$ . Média final: 0.8007