Intervalo de Confiança para o Tempo de Execução

Coloque seu nome aqui

Data de entrega: 07/05/2021

Descrição da atividade

O objetivo desta atividade é aplicar o conceito de intervalo de confiança à medição do tempo de execução de uma função.

Algumas recomendações:

- Se você não estiver habituado com R Markdown, acostume-se a processar com frequência o documento, usando o botão **Knit**. Isso permitirá que eventuais erros no documento ou no código R sejam identificados rapidamente, pouco depois de terem sido cometidos, o que facilitará sua correção. Na verdade, é uma boa ideia você fazer isso **agora**, para garantir que seu ambiente esteja configurado corretamente. Se você receber uma mensagem de erro do tipo *Error in library(foo)*, isso significa que o pacote `foo` não está instalado. Para instalar um pacote, execute o comando `install.packages("foo")` no Console, ou clique em *Tools -> Install Packages*.
- Após concluir a atividade, você deverá submeter no Moodle um arquivo ZIP contendo:
  - o arquivo fonte .Rmd;
  - a saída processada (PDF ou HTML) do arquivo .Rmd;
  - o(s) arquivo(s) de dados necessário(s) para o processamento do .Rmd, caso você opte por salvar os dados analisados separadamente.

Configuração

Nesta atividade, a única configuração necessária consiste em carregar o arquivo `rand171.R`, que contém a função `rand171()`, cujo tempo de execução será medido.

```
source("rand171.R")          # carrega arquivo que contem a funcao
```

Salvando seus dados

Este é um experimento iterativo com dados que não se repetem: você irá realizar uma primeira rodada de execuções, analisar os dados dessa rodada para determinar quantas execuções adicionais serão necessárias, e analisar os dados da segunda rodada para saber se serão necessárias mais execuções. Nesse processo, você irá trabalhando com os dados à medida em que eles são obtidos. Ao final do processo, você precisa:

- salvar seus dados em um arquivo;
- escrever sua análise dos dados com base nos dados salvos.

O salvamento dos dados é importante porque eles estarão disponíveis no seu ambiente, mas não para quem receber o seu arquivo .Rmd. Assim, os dados precisam ser fornecidos de alguma forma para que seja possível reprocessar o arquivo .Rmd e reproduzir a análise usando os dados originais.

Você pode salvar seus dados de duas formas. A primeira é preservá-los no próprio arquivo .Rmd, e a segunda é usando um arquivo de dados auxiliar.

Salvando os dados no próprio arquivo .Rmd

Supondo que seus tempos de execução estejam armazenados em um vetor `texec`, você pode usar a função `dput()` para mostrar o conteúdo do vetor de modo que possa ser atribuído a uma variável. O código abaixo demonstra isso com um vetor `x`:

```
x <- 2^c(1:8)
dput(x)
```

```
## c(2, 4, 8, 16, 32, 64, 128, 256)
```

Veja que `dput()` enumera os elementos do vetor em um formato que permite a atribuição a uma variável: os valores de `x` poderiam ser preservados colocando `x <- c(2, 4, 8, 16, 32, 64, 128, 256)` em um bloco R no arquivo .Rmd. Você pode usar a mesma estratégia para preservar os valores de `texec`.

Salvando os dados em um arquivo auxiliar

Supondo novamente que seus tempos de execução estejam armazenados em um vetor `texec`, o código abaixo mostra como salvá-los em um arquivo `texec-rand171.dat`.

```
# texec contém os tempos de execução medidos no experimento
df <- data.frame(texec)
write.table(df, "texec-rand171.dat", row.names = FALSE, quote = FALSE)
```

O código abaixo pode ser usado para recuperar `texec` a partir do arquivo salvo pelo código anterior:

```
texec <- read.table("texec-rand171.dat", header = TRUE)$texec
```

Caso você opte por salvar os dados em um arquivo separado, o comando que lê os dados do arquivo deve ser incluído no seu bloco R de análise.

Definição do experimento

A função `rand171()` encontra `noc` ocorrências do número 171 em uma sequência de números aleatórios, e imprime e retorna o seu tempo de execução (em segundos).

1. Determine o número n de execuções necessárias para obter, com 95% de confiança, o tempo médio de execução com uma margem de erro de ±6%. Considere uma amostra inicial de 5 execuções. Se necessário, ajuste o parâmetro `noc` para que o tempo de execução fique na ordem de 3 a 5 s (o valor *default* é 20, que deve ser adequado para o RStudio Cloud). Para obter n, use a equação 4.21 (pág. 53) do livro; adote $e = 0.06$, e lembre-se que o coeficiente z deve ser obtido da distribuição t (pois a amostra inicial tem 5 elementos).
2. Realize o número de execuções determinadas no passo anterior, e calcule o intervalo de confiança de 95% para a média. Observe que você deve considerar as 5 execuções iniciais: por exemplo, se no passo 1 você descobriu que são necessárias 25 execuções, execute a função mais 20 vezes, e calcule o IC usando todas as 25 execuções.
3. O IC obtido no item anterior fica dentro da margem de erro desejada de ±6%? Caso contrário, amplie a amostra até obter a margem pretendida, e informe o novo IC calculado e quantas execuções foram necessárias ao final. A margem de erro é dada por

$$e = \frac{zs}{\bar{x}\sqrt{n}}$$

onde o coeficiente z deve ser obtido da distribuição t (se $n < 30$) ou da distribuição normal (se $n \geq 30$). Para obter uma porcentagem, multiplique e por 100.

4. Compare os histogramas dos tempos de execução obtidos nos passos 1, 2 e 3 (se houver).

Respostas

Como explicado acima, você deve salvar os dados obtidos durante a realização do experimento e depois escrever sua análise com base nos dados salvos. *Trabalhe com os tempos já coletados, não é necessário realizar novas execuções de* `rand171()`.

A forma mais fácil de fazer isso é usando um bloco separado para cada uma das questões 1 a 4. Você pode adaptar a estrutura abaixo para isso.

Questão 1

```
# código para analisar os dados da questão 1 aqui
```

Resposta para a questão 1

Questão 2

```
# código para analisar os dados da questão 2 aqui
```

Resposta para a questão 2

...