

Comparação de Alternativas

Coloque seu nome aqui

Data de entrega: 19/04/2024

Descrição da atividade

O objetivo desta atividade é aplicar as técnicas de comparação de alternativas. A atividade é dividida em duas partes:

1. Comparação usando ICs vs. teste t
2. Comparação de múltiplas alternativas

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 arquivo de dados referente à Parte 2, que é necessário para o processamento do `.Rmd`.

Configuração

Nesta atividade, a única configuração necessária consiste em carregar o pacote `ggplot2` e o arquivo `compar-altern.R`, que são usados na Parte 1 da atividade.

```
library(ggplot2)
source("compar-altern.R")
```

Parte 1: Comparação usando ICs vs. teste t

Uma das formas de determinar se duas variáveis são estatisticamente diferentes é observando os seus intervalos de confiança. Existem três resultados possíveis para essa comparação:

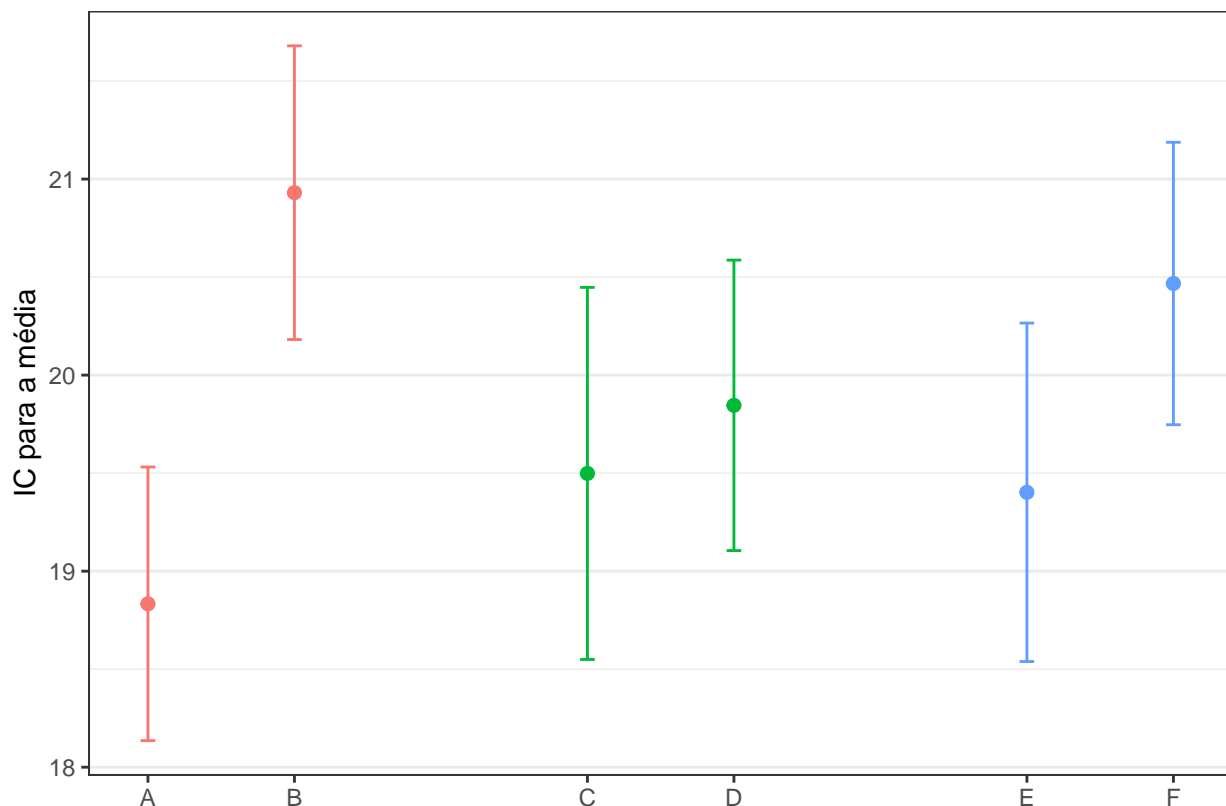
1. *Não existe sobreposição entre os ICs*. Nesse caso, a diferença entre as variáveis **é** estatisticamente significativa.
2. *Existe sobreposição entre os ICs, e ao menos um deles inclui a média da outra variável*. Nesse caso, a diferença entre as variáveis **não é** estatisticamente significativa.

3. *Existe sobreposição entre os ICs, mas nenhum deles inclui a média da outra variável.* Nesse caso não é possível afirmar nada, sendo necessário realizar um teste t (ou equivalente) para determinar se a diferença é estatisticamente significativa.

O gráfico abaixo ilustra os três resultados. As variáveis comparadas são as colunas A–F do conjunto de dados contido no arquivo `comparacao-ic.dat`, e os ICs têm um nível de confiança de 95%. As conclusões visuais são as seguintes:

1. As variáveis A e B possuem diferença estatisticamente significativa, e $A < B$.
2. As variáveis C e D não possuem diferença estatisticamente significativa.
3. Não é possível afirmar se $E < F$ ou não, é preciso realizar um teste t para ver se a diferença é estatisticamente significativa.

```
dados <- read.table("comparacao-ic.dat", head=TRUE)
dados.ic <- geraIC(dados)
plotaIC(dados.ic)
```



Para esta primeira parte, você deve comparar os pares de variáveis representados no gráfico (A/B, C/D, E/F) usando o teste t com um nível de confiança de 95% (o mesmo usado para gerar os ICs). Para cada par de variáveis, indique claramente (a) o resultado da comparação (ou seja, se as variáveis são ou não estatisticamente diferentes) e (b) se esse resultado é idêntico ao obtido pela comparação visual dos ICs. Considere que as observações não são pareadas.

Análise e respostas

seu código R aqui

Respostas aqui

Parte 2: Comparação de três algoritmos de ordenação

Na segunda parte iremos comparar o tempo de execução de três algoritmos de ordenação, *QuickSort*, *MergeSort* e *HeapSort*. Esses três algoritmos têm complexidade $O(n \log n)$ no caso médio, e são considerados eficientes. Para essa comparação iremos usar tempos de execução medidos pelo script Python `sortcomp3.py`. Esse script mede o tempo que cada algoritmo leva para ordenar um vetor de n elementos (em uma rodada, cada algoritmo ordena um vetor diferente, sempre de tamanho n). O número de rodadas pode ser passado como parâmetro na linha de comando (por *default* são realizadas 3 rodadas). A cada rodada os elementos do vetor sofrem uma permutação aleatória; logo, é possível (mas pouco provável) que o vetor esteja (quase) em ordem (de)crescente.

O script pode ser executado no RStudio, incluindo a versão Cloud. Na janela inferior esquerda, normalmente usada para o console, há uma aba Terminal, na qual você pode executar comandos do Linux.

Neste experimento, primeiro execute o script usando o comando `python sortcomp3.py 2`. O número de rodadas (2, no exemplo) fica a seu critério.

A seguir, faça uma análise de variância adotando um nível de confiança de 95%, e responda aos seguintes itens:

1. Qual a porcentagem de variação que pode ser explicada pelas alternativas e qual a porcentagem explicada pelo ruído das medições?
2. Mostre a tabela ANOVA (conforme o esquema abaixo) e determine se existem diferenças estatisticamente significativas entre os tempos médios de resposta de cada algoritmo.

Fonte de variação	Alternativas	Erros	Total
Soma de quadrados			
Graus de liberdade			
Média quadrática			
F calculado			
F crítico			

3. Caso a ANOVA indique que há diferenças estatisticamente significativas, ranqueie os algoritmos de acordo com o seu tempo médio de resposta (use o teste de Tukey).

Lembre-se que os tempos de execução dos algoritmos devem ser salvos em um arquivo de dados para que sua análise seja reproduzível. Para facilitar essa tarefa, o script já gera a saída em um formato apropriado; você pode redirecionar a saída do script para um arquivo (por exemplo, `python sortcomp3.py 2 >parte2.dat`) ou simplesmente criar o arquivo de dados no próprio editor do RStudio (crie um novo arquivo texto e cole a saída do script).

Análise e respostas

seu código R aqui

Respostas aqui