



Universidade Federal  
de São João del-Rei

Aluno: **Diego Arthur Bispo Justino de Oliveira**  
Orientador: **Dr. Ben Dêvide de Oliveira Batista**

## **RELATÓRIO**

**MCPtests: Um pacote R para Procedimentos de comparações  
múltiplas**

**OURO BRANCO - MG**

**2020**

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>2</b>
<b>2</b>	<b>REVISÃO DE LITERATURA</b>	<b>3</b>
<b>3</b>	<b>METODOLOGIA</b>	<b>7</b>
<b>3.1</b>	<b>Uso do pacote MCPtests</b>	<b>7</b>
<b>3.1.1</b>	<b>Função MCPtest</b>	<b>8</b>
<b>3.1.2</b>	<b>Função MCPbarplot</b>	<b>8</b>
<b>3.1.3</b>	<b>Função MCPwrite</b>	<b>9</b>
<b>3.2</b>	<b>Interface do pacote MCPTests</b>	<b>9</b>
<b>3.2.1</b>	<b>Entrada conjunto de dados e outras funções</b>	<b>9</b>
<b>3.2.2</b>	<b>Testes e Extensão</b>	<b>10</b>
<b>3.2.3</b>	<b>Opções de entrada de dados e Nível de significância</b>	<b>10</b>
<b>3.2.4</b>	<b>Console</b>	<b>10</b>
<b>3.2.5</b>	<b>Gráfico</b>	<b>10</b>
<b>3.2.6</b>	<b>Parâmetros Gráficos</b>	<b>10</b>
<b>3.2.7</b>	<b>Botão I (Informação)</b>	<b>11</b>
<b>3.2.8</b>	<b>Aplicações do Pacote MCPtests</b>	<b>11</b>
<b>4</b>	<b>RESULTADOS E DISCUSSÃO</b>	<b>12</b>
<b>4.1</b>	<b>Usando o pacote MCPtests</b>	<b>12</b>
<b>4.1.1</b>	<b>Opção Modelo</b>	<b>12</b>
<b>4.1.2</b>	<b>Opção Variável Resposta</b>	<b>13</b>
<b>4.1.3</b>	<b>Opção Média</b>	<b>14</b>
<b>5</b>	<b>CONCLUSÃO</b>	<b>17</b>
	<b>REFERÊNCIAS</b>	<b>18</b>

## 1 INTRODUÇÃO

Com o avanço tecnológico e dos computadores, o auxílio da estatística passou a estar cada vez mais presente na ciência. Um dos fatores, foi a utilização de métodos estatísticos mais complexos para a solução de problemas mais diversos possíveis. Restrito ao desenvolvimento de softwares estatísticos, em 1995, Robert Gentleman e Ross Ihaka desenvolveram a primeira versão da linguagem R R CORE TEAM (2020) para desenvolvimento de problemas estatísticos e gráficos. O que foi mais inovador para a comunidade estatística foi uma linguagem de código livre e orientada a objetos. Outra vantagem é a versatilidade que essa linguagem tem com outras linguagens, como FORTRAN, Python, C, C++, Java, Julia, dentre outras. Isso proporcionou uma ampla e alta utilização do R para uso de análises estatísticas na comunidade científica. Contudo, uma de suas desvantagens para áreas aplicadas é o desenvolvimento das rotinas para análise de dados, que muitos usuários sentem dificuldade, uma vez, que exige um certo conhecimento sobre programação.

Este trabalho tem como objetivo revisar alguns dos PCMs desenvolvidos até o momento na literatura e implementá-los no pacote MCPtests. O pacote terá uma interface gráfica para facilitar a acessibilidade e utilização. O *help* do pacote, fornece um resumo de como utilizar as funções para a execução dos testes. Portanto, os objetivos desse trabalho são: revisar alguns dos principais procedimentos de comparações múltiplas desenvolvidos até o momento e implementá-los no pacote MCPtests.

## 2 REVISÃO DE LITERATURA

Ao se testar uma hipótese nula de igualdade de duas ou mais médias, duas decisões podem ser tomadas: (i) não rejeitar a hipótese ou (ii) rejeitá-la. Considerando a segunda situação, fica uma outra indagação: quais médias diferem entre si? Vários pesquisadores, tais como Hsu (1996), Hochberg e Tamhane (1987), Benjamini, Bretz e Sarkar (2004), Bretz, Hothorn e Westfall (2011), Dickhaus (2014), Gelman, Hill e Yajima (2012), Miller (1981), Batista e Ferreira (2020), entre outros, têm dado atenção a essa questão, propondo uma variedade de procedimentos de comparações múltiplas (PCMs), que identificam as diferenças existentes entre as médias.

Como há uma enorme quantidade de procedimentos desenvolvidos na literatura, vamos apresentar para esse relatório, apenas alguns procedimentos baseados na distribuição da amplitude *estudentizada* externamente. É bom lembrar, que muitas outras metodologias de procedimentos também estão inseridos no pacote, e que esse projeto mesmo sendo finalizado, o pacote `MCPtests` sempre estará em atualização, e mais procedimentos serão inseridos.

Antes de mostrar os testes de comparações múltiplas baseados nessas distribuições, é interessante entender como esses métodos são desenvolvidos. Um dos interesse dos PCMs está em verificar um conjunto de parâmetros originados da combinação linear de médias tomadas duas a duas, expressos por  $\theta_l = \mu_i - \mu_{i'}$ ,  $i \neq i' \in \{1, 2, \dots, n\}$ , para  $l = 1, 2, \dots, N$ . A coleção desses parâmetros é chamado de família (HOCHBERG; TAMHANE, 1987), sendo o número de parâmetros dado por

$$N = \binom{n}{2} = \frac{n(n-1)}{2}, \quad (1)$$

considerando as diferenças das  $n$  médias duas a duas.

Cada parâmetro é avaliado por meio de um teste de hipóteses. E cada teste está associado erros de decisão: erro tipo I e erro tipo II. Assim, o desenvolvimento de um PCM é realizado avaliando o seu desempenho por meio do erro tipo I e o poder, sendo este último o complemento do erro tipo II. A Tabela 1 mostra o resumo dos tipos de erros envolvidos numa tomada de decisão sob a hipótese nula  $H_0$  e nas respectivas probabilidades. Para avaliar os erros em um determinado teste, é realizado cenários experimentais simulados baseados no método Monte Carlo. Por exemplo, para uma avaliação do erro tipo I, a simulação realizada supõe hipótese nula global ( $H_0$ ), isto é, a simulação realizada gera amostras de  $n$  tratamentos em um experimento da mesma população, portanto, com mesma média populacional. Para um maior detalhamento, uma revisão sobre o desenvolvimento das distribuições da amplitude *midrange* *estudentizada* externamente, bem como dos procedimentos de comparações múltiplas estão bem documentadas em Batista (2016).

Tabela 1 Tipos de erros e decisões corretas, com suas respectivas probabilidades em um teste de hipótese.

Decisão	Realidade	
	$H_0$ verdadeira	$H_0$ falsa
Não Rejeitar $H_0$	Decisão correta $1 - \alpha$	Erro Tipo II $\beta$
Rejeitar $H_0$	Erro tipo I $\alpha$	Poder $1 - \beta$

Os testes baseados no quantil superior  $100\alpha\%$  da distribuição da amplitude estudentizada externamente, denotado por  $q_{(\alpha;n,\nu)}$ , com  $n$  tratamentos e  $\nu$  graus de liberdade do resíduo serão apresentados na sequência. Os valores de  $q_{(\alpha;n,\nu)}$ , podem ser obtidos pela função `qtukey()` (R CORE TEAM, 2020), tais que:

$$P [Q \leq q_{(\alpha;n,\nu)}] = 1 - \alpha. \quad (2)$$

Considere a seguinte amostra aleatória  $Y_{11}, Y_{12}, \dots, Y_{1r}, Y_{21}, \dots, Y_{2r}, \dots, Y_{i1}, Y_{i2}, \dots, Y_{ij}, \dots, Y_{ir}, \dots, Y_{n1}, Y_{n2}, \dots, Y_{nr}$ , sendo  $Y_{ij}$  a observação aleatória referente ao  $i$ -ésimo tratamento e a sua  $j$ -ésima repetição,  $i = 1, 2, \dots, n$  e  $j = 1, 2, \dots, r$ , em que a média do  $i$ -ésimo tratamento é dada por:

$$\bar{Y}_i = \frac{\sum_{j=1}^r Y_{ij}}{r} = \frac{Y_{i.}}{r}, \quad (3)$$

sendo a variável aleatória  $\bar{Y}_i$  é normalmente distribuída, independente e homocedástica (variância  $\sigma^2$ ), então a estatística  $Q$ , pode ser definida como

$$Q = \frac{\bar{Y}_{(n)} - \bar{Y}_{(1)}}{S_{\bar{Y}_i}}, \quad (4)$$

em que  $\bar{Y}_{(n)}$  e  $\bar{Y}_{(1)}$  são as estatísticas de ordem, referentes a maior e a menor média em  $n$  tratamentos, respectivamente;  $S_{\bar{Y}_i} = \sqrt{QME/r}$  é o erro padrão para uma média amostral; e  $QME$  é o quadrado médio do resíduo.

Os testes de comparações múltiplas baseados na estatística  $Q$  comparam todas as médias duas a duas, ou seja, testa a hipótese  $H_0 : \mu_i - \mu_{i'} = 0$  para  $i \neq i' \in \{1, 2, \dots, n\}$ . A decisão do teste é baseada na diferença mínima significativa ( $\Delta$ ) ao nível de significância  $\alpha$ , que será definida posteriormente para cada teste apresentado. Assim, todos os contrastes  $D = \bar{Y}_i - \bar{Y}_{i'}$  que superam o valor de  $\Delta$ , serão considerados significativamente diferentes de zero, e a hipótese  $H_0 : \mu_i - \mu_{i'} = 0$  deve ser rejeitada naquele nível nominal de significância  $\alpha$  estabelecido. O primeiro teste, que deu início a uma sequência de outros dois, é o teste Tukey (TUKEY, 1953).

**Proposição 2.1 — Teste de Tukey (1953).** Considere  $n$  amostras normalmente distribuídas, independentes e homocedásticas, de tamanho  $r$ , e  $\bar{Y}_i$ , a média da  $i$ -ésima amostra (tratamento),  $i = 1, 2, \dots, n$ . Então a hipótese  $H_0 : \mu_i - \mu_{i'} = 0$  para  $i \neq i' \in \{1, 2, \dots, n\}$ , será testada comparando  $|\bar{Y}_i - \bar{Y}_{i'}|$  com  $\Delta = q_{(\alpha;n,\nu)} \sqrt{\frac{QME}{r}}$ , em que  $q_{(\alpha;n,\nu)}$  é o quantil superior a  $100\alpha\%$  da distribuição da amplitude estudentizada externamente. Se

$$|\bar{Y}_i - \bar{Y}_{i'}| \geq q_{(\alpha;n,\nu)} \sqrt{\frac{QME}{r}}, \quad (5)$$

a hipótese  $H_0$  é rejeitada, isto é,  $\mu_i$  e  $\mu_{i'}$  são consideradas significativamente diferentes de zero, no nível de significância  $\alpha$ . ■

O segundo teste de SNK foi desenvolvido por Student (1927), Newman (1939) e Keuls (1952). Este teste é similar ao teste de Tukey, com uma pequena diferença que leva em consideração o número de médias abrangidas no contraste entre médias ordenadas.

**Proposição 2.2 — Teste de Student-Newman-Keuls (SNK).** Considere  $n$  amostras normalmente distribuídas, independentes e homocedásticas, de tamanho  $r$ , e  $\bar{Y}_i$ , a média da  $i$ -ésima amostra (tratamento),  $i = 1, 2, \dots, n$ . Então a hipótese  $H_0 : \mu_i - \mu_{i'} = 0$  para  $i \neq i' \in \{1, 2, \dots, n\}$ , será testada comparando  $|\bar{Y}_i - \bar{Y}_{i'}|$  com  $\Delta = q_{(\alpha;p,\nu)} \sqrt{\frac{QME}{r}}$ , em que  $q_{(\alpha;p,\nu)}$  é o quantil superior a  $100\alpha\%$  da distribuição da amplitude estudentizada externamente, sendo  $p = 2, 3, \dots, n$ . Nesse caso, as médias devem ser ordenadas em primeiro lugar. Se

$$|\bar{Y}_i - \bar{Y}_{i'}| \geq q_{(\alpha;p,\nu)} \sqrt{\frac{QME}{r}}, \quad (6)$$

a hipótese  $H_0$  é rejeitada, isto é,  $\mu_i$  e  $\mu_{i'}$  são consideradas estatisticamente diferentes, no nível nominal de significância  $\alpha$ . ■

O terceiro teste abordado baseado na amplitude estudentizada externamente, mas que não fará parte dos estudos desse trabalho, apresentado apenas pela sua importância, é o teste de Duncan. Esse teste sofre duras críticas, pois apresenta alta taxa de erro tipo I por experimento (O'NEILL; WETHERILL, 1971).

**Proposição 2.3 — Teste de Duncan (1955).** Considere  $n$  amostras normalmente distribuídas, independentes e homocedásticas, de tamanho  $r$ , e  $\bar{Y}_i$ , a média da  $i$ -ésima amostra (tratamento),  $i = 1, 2, \dots, n$ . Então a hipótese  $H_0 : \mu_i - \mu_{i'} = 0$  para  $i \neq i' \in \{1, 2, \dots, n\}$ , será testada comparando  $|\bar{Y}_i - \bar{Y}_{i'}|$  com  $\Delta = q_{(\alpha_p;p,\nu)} \sqrt{\frac{QME}{r}}$ , em que  $q_{(\alpha_p;p,\nu)}$  é o quantil superior a  $100\alpha_p\%$  da distribuição da amplitude estudentizada externamente, expressão (2), e  $\alpha_p = 1 - (1 - \alpha)^{p-1}$ , sendo  $p = 2, 3, \dots, n$ . Se

$$|\bar{Y}_i - \bar{Y}_{i'}| \geq q_{(\alpha_p;p,\nu)} \sqrt{\frac{QME}{r}}, \quad (7)$$

a hipótese  $H_0$  é rejeitada, isto é,  $\mu_i$  e  $\mu_{i'}$  são consideradas estatisticamente diferentes, no nível nominal de significância  $\alpha$ . ■

O teste de Duncan tem uma pequena modificação quanto ao teste de SNK. No cálculo do quantil da distribuição da amplitude estudentizada externamente, o nível de significância é variável de acordo com o número de médias abrangidas, proporcionando um total de  $(n - 1)$  valores críticos. Além disso, o valor do quantil da amplitude estudentizada externamente varia conforme o número de  $p$  médias abrangidas pelo contraste.

Como já falado anteriormente, a implementação desses testes, dentre outros, foram implementados no pacote `MCPtests`, usando a linguagem R. A linguagem R tem uma estrutura que denominamos de pacote. O pacote é um conjunto de funções programadas para resolver um determinado problema. Quando se instala o R, existe na instalação um conjunto de pacotes básicos para resolver uma gama de problemas estatísticos. Entretanto, caso deseje contribuir para a linguagem, podem ser desenvolvidos pacotes e estes serem enviados para a plataforma do R, chamada CRAN. Caso algum usuário esteja interessado no pacote, este pode ser

baixado e utilizado. É assim, que toda comunidade científica contribui para o R.

Alguns pacotes para utilização de procedimentos de comparações múltiplas já existem, tais como `multcomp` Hothorn, Bretz e Westfall (2008), `agricolae` Mendiburu e Mendiburu (2020), `ExpDes` Ferreira, Cavalcanti e Nogueira (2018), entre outros. Contudo, esses pacotes têm uma limitação do número de comparações múltiplas disponíveis, e ainda mais, não há alguma referência sobre as características de avaliação de desempenho dos testes. Isso torna um ponto no escuro para o usuário, em achar que poderá utilizar qualquer PCM. Como conclusão, o pesquisador poderá tomar decisões equivocadas. Não se pode deixar de mencionar que existem alguns programas estatísticos de alto potencial de análise de dados, tais como SAS, Statistica, Minitab, Sisvar, etc. Contudo, esses programas são implementados em códigos fechados, isto é, o usuário não tem acesso a estrutura interna de programação. Dessa forma, problemas do tipo: rotinas programadas com base em teorias equivocadas, análise não implementadas, dentre outras, podem tornar uma limitação para utilização do software, pois não saberemos como as metodologias estão sendo implementadas.

Com isso, implementamos o pacote `MCPtests`, em uma linguagem de código aberto (linguagem R), com uma versatilidade para o usuário com conhecimento na linguagem ou não. Para isso, também desenvolvemos uma interface gráfica para o usuário (IGU), usando a linguagem Tcl/Tk, por meio do pacote `tcltk`, da base do R. Outras vantagens do pacote, estão nas entradas de dados, do qual apresentamos três formas, algo não apresentado nos outros pacotes bem como softwares. Por fim, as saídas gráficas dos resultados dos testes, bem como a exportação destes em diversos formatos, podem ser obtidos. Detalhes serão descritos a seguir.

### 3 METODOLOGIA

A fase inicial do projeto, apesar da revisão de literatura dos testes, o que nos demandou um maior consumo de tempo foi a IGU do pacote. Algoritmos iniciais podem ser apresentados, como ilustração de alguns elementos gráficos dessa interface. Nesse primeiro algoritmo, apresentamos a formação da janela principal onde será inserido os componentes ou *widgets* (elementos gráficos).

---

#### Algorithm 1: Janela Principal ( `tkoplevel` )

---

```
1: mcpprincipal <- tkoplevel(
2: width = 800,
3: height = 600
4: )
```

---

O algoritmo seguinte mostra um elemento gráfico muito interessante que é o *panedwindow*, uma espécie de *frame* flexível, isto é, o usuário pode redimensionar as dimensões dos quadrantes da interface. Essa função dá a movimentação em sentidos sul, norte, leste e oeste.

---

#### Algorithm 2: Widget flexível ( `ttkpanedwindow` ( ) )

---

```
1: group.all <- ttkpanedwindow(mcpprincipal, orient = "horizontal")
2: tkpack(group.all, expand = TRUE, fill = "both")
```

---

Um outro elemento gráfico interessante na interface é um organizador de elementos gráficos, também chamado de *frame*. O objetivo é organizar outros elementos gráficos como botões, rótulos, caixas de entradas, etc.

---

#### Algorithm 3: Widget Frame ( `tkframe` )

---

```
1: tkpack(fentry <- tkframe(child1.group1, relief = "sunken"), anchor = "n",
2: expand = TRUE, fill = "x")
```

---

Poderíamos mencionar aqui muitos outros elementos gráficos utilizados na IGU do pacote `MCPtests`. Contudo, nos limitamos a estes por simples ilustração de como demanda tempo um desenho de uma interface. Nos passos seguintes, demonstraremos como utilizar o pacote em sua versão mais ampla, pois caso o usuário não deseje utilizar a interface, pode também ser utilizado linhas de códigos. Para isso, vejamos as funções implementadas no pacote `MCPtests`.

#### 3.1 Uso do pacote `MCPtests`

Para instalar o pacote basta entrar com os seguintes comandos no console do R:

```
install.packages ("MCPtests") # Instalando o pacote
library("MCPtests") # Carregando o pacote
```



O pacote apresenta três funções básicas, `MCPtest`, `MCPbarplot` e `MCPwrite`. Vejamos detalhadamente, a finalidade de cada argumento dessas funções.

### 3.1.1 Função `MCPtest`

A função `MCPtest` é expressa da seguinte forma:

```
MCPtest (y, trt = NULL, dferror = NUL, merror = NULL,
         replication = NULL, alpha = 0.05, main = NULL,
         MCP = "all", ismean = FALSE)
```

sendo:

- `y`: vetor com as variáveis respostas, o modelo desejado ou a média dos tratamentos;
- `trt`: vetor com os níveis do fator em estudo;
- `dferror`: grau de liberdade do resíduo;
- `msferror`: quadrado médio do resíduo;
- `alpha`: nível de significância adotado nos testes;
- `replication`: número de repetições dos tratamentos;
- `main`: título de sua análise;
- `MCP`: permite escolher o teste de comparação múltipla desejado;
- `ismean`: argumento lógico. Se `ismean = TRUE`, é porque o argumento `y` é um vetor de médias dos tratamentos. Caso contrário, `ismean = FALSE`.

Essa função é a principal do pacote. Por meio dela, obtemos todos os resultados dos testes em seu detalhamento, e também, do qual as outras funções precisarão dessas informações em seus argumentos. A próxima função, apresentará em forma de gráfico os resultados obtidos por `MCPtest`.

### 3.1.2 Função `MCPbarplot`

A função `MCPbarplot` é expressa da seguinte forma:

```
MCPbarplot(x, MCP = "all", col = heat.colors(10), horiz = FALSE, ...)
```

sendo:

- `x`: objeto da função `MCPtest`;
- `MCP`: permite escolher o teste de comparação múltipla desejado;
- `col`: permite escolher a cor das barras do gráfico;
- `horiz`: escolhe a direção das barras.
- ... a função permite que o usuário possa acrescentar mais argumentos da função `barplot`.

Por fim, a parte de exportação do resultados dados por `MCPtest`, podem ser realizados pela função `MCPwrite`, apresentada a seguir.

### 3.1.3 Função `MCPwrite`

A função `MCPwrite` é expressa da seguinte forma:

```
MCPwrite(x, MCP = "all", extension = "csv", dataMCP = "all")
```

sendo:

- `x`: objeto da função `MCPtest`;
- `MCP`: permite escolher o teste de comparação múltipla desejado;
- `extension`: tipo de formato de arquivo;
- `dataMCP`: permite escolher os resultados a serem exportados.

## 3.2 Interface do pacote `MCPTests`

Apresentaremos nessa seção, como a IGU pode ser utilizada no pacote `MCPtests`

### 3.2.1 Entrada conjunto de dados e outras funções

Estas funções se encontram na parte de Menu da interface. Este menu possui as opções "File" e "Edit". Na opção "File" encontra-se as opções de "Escolha do diretório" e "Entrada de dados". Já no "Edit" há a opção "edição do conjunto de dados".

File → Choose directoy: o primeiro passo para a análise dos dados é a escolha do diretório(pasta) para que o conjunto de dados analisados seja exportado para este local, sendo ele de variados tipos de extensões que apresentaremos aqui.

File → Open file (.txt or .csv): o segundo passo é a seleção do tipo de separador de variáveis do seu conjunto de dados. Exemplo: caso as suas variáveis do conjunto seja separado por "," ou ";" terá que ser determinado, caso contrário, o programa não reconhecerá. O sistema de norma internacional utiliza o "." como separador padrão, então, neste caso, não terá a necessidade da escolha do tipo.

File → Open file (.txt or .csv) → Enter the data : o passo seguinte é a importação do conjunto de dados que podem ser do tipo .txt ou .csv.

Uma vez importado, para que o seu conjunto de dados esteja correto, terá que aparecer na primeira linha os nomes das variáveis que foram atribuídos para cada coluna.

Edit → Data set...: caso haja a necessidade de edição de algum dado, a opção "Edit/Data set" proporcionará isso. Mas atenção, esta edição só será feita dentro do programa e não no arquivo de origem.

Aqui é apresentado o caminho da pasta de origem de onde o conjunto de dados foi importado, a discriminação das colunas do conjunto por meio dos nomes dados dentro do arquivo.

### 3.2.2 Testes e Extensão

Antes da entrada de dados essenciais para a análise do conjunto, é determinado o tipo de teste que será realizado. O tipo da extensão para a exportação do conjunto de dados analisados tem que ser definido também, sendo eles 'txt', 'xlsx', 'csv' e 'latex'. Ressaltando que a extensão do tipo 'latex' é o único que não é direcionado para o diretório escolhido, pois é mostrado na tela do console que será explicado em breve.

### 3.2.3 Opções de entrada de dados e Nível de significância

O nível de significância da análise dos dados tem que ser definido, mas por padrão é 0,05. As opções de entrada de dados é a que determina a forma como será calculado o teste escolhido e, por ter mais escolhas de entrada, é o grande diferencial deste programa para outros existentes, inclusive softwares. A primeira opção denominada como 'Modelo' é a mais comum, pois todos os dados para a análise devem ser fornecidas, sendo elas contidas no arquivo. As outras opções de entrada como a 'Variável Resposta' e 'Médias' traz uma possibilidade de proteção aos dados, caso a seguinte situação seja de não querer fornecer o conhecimento do Modelo do conjunto, mas a de ter algumas e principais informações do experimento. Para isso, nessa última opção o usuário pode ter como entrada apenas as médias dos tratamentos, quadrado médio e grau de liberdade do resíduo do experimento.

### 3.2.4 Console

Após clicar o botão Calcular, os dados de análise do conjunto serão mostrados no Console. Serão exibidos o sumário, o resultado do teste, a média do experimento, o coeficiente de variação, o quadrado médio do resíduo, os graus de liberdade e, caso a Extensão de Saída esteja como latex, o código do mesmo será exibido também no console.

### 3.2.5 Gráfico

O gráfico da análise é exibido aqui. Na barra que está acima do gráfico, é a configuração de escala, ou seja, o tamanho que o mesmo é apresentado. A opção "Salvar como..."vem com opções de salvar a imagem com tipos variados como "pdf", "png", "jpeg"... Por default a imagem da análise é preto e branco, mas podendo ser alterado no espaço de Parâmetros Gráficos.

### 3.2.6 Parâmetros Gráficos

Esta função traz a possibilidade de atualização de três componentes relacionados ao gráfico. O primeiro componente é a mudança de cor que pode ser mudada. Já o segundo componente é a mudança de sentido dos eixos do gráfico sendo ela na Vertical (padrão) ou Horizontal. O terceiro componente é a opção de nomear os eixos do gráfico.

### 3.2.7 Botão I (Informação)

O botão I (Informação) é para auxiliar com exemplos e explicações seja sobre os Testes, Tipo de entrada de dados, Extensão, Nível de significância, dentre outras inserções de dados dentro do programa.

### 3.2.8 Aplicações do Pacote MCPtests

A aplicação do pacote MCPtests será feita utilizando uma aplicação similar ao que foi encontrado em Batista (2016). O experimento foi retirado de Steel e Torrie (1980, p. 180) que representará o Exemplo 3.1.

■ **Exemplo 3.1** O experimento foi realizado no Delineamento Inteiramento Causalizado (DIC) e objetiva avaliar o efeito das bactérias no teor de nitrogênio das plantas de trevo vermelho. Esse experimento apresentou os seguintes dados:

Tabela 2 Dados de teor de nitrogênio das plantas de trevo vermelho relacionados as bactérias em estudo.

Tipo de Bactéria	Repetições				
	$R_1$	$R_2$	$R_3$	$R_4$	$R_5$
3DOK1	19,4	32,6	27,0	32,1	33,0
3DOK5	17,7	24,8	27,9	25,2	24,3
3DOK4	17,0	19,4	9,1	11,9	15,8
3DOK7	20,7	21,0	20,5	18,8	18,6
3DOK13	14,3	14,4	11,8	11,6	14,2
COMPOS	17,3	19,4	19,1	16,9	20,8

Para esse exemplo, será mostrado duas formas de entrada de dados para a execução da função MRtest. A primeira, o argumento  $y$  da função MRtest recebendo o objeto que armazenou o resultado da função aov(), e o segundo o argumento  $y$  recebendo o vetor da variável resposta (teor de nitrogênio). Posteriormente, recursos gráficos serão apresentados, bem como a exportação dos resultados dos testes executados.

## 4 RESULTADOS E DISCUSSÃO

A seguir, vamos apresentar a aplicação do pacote `MCPtests` em suas duas formas: linha de código e IGU. Faremos a ilustração de apenas um teste aplicado, porém, muitos outros podem ser utilizados.

### 4.1 Usando o pacote `MCPtests`

A primeira função do pacote `MCPtests` a ser apresentada é o `MCPtest`. Inicialmente, **foi criado criamos** um arquivo que armazenou o banco de dados do Exemplo 3.1, intitulado `datast1980.txt`. O tipo de teste utilizado para análise dos dados aqui é o `SNKM`, não apresentado na seção de referencial teórico, mas muito bem documentada em Batista (2016).

#### 4.1.1 Opção Modelo

Na opção Modelo **são inseridos inserimos** o modelo( $y \sim \text{trt}$ ) e o tratamento(`trt`):

```
dados <- read.table("datast1980.txt", h =T)
dados$trt <- factor(dados$trt)
attach(dados) # Quebrando o objeto dados
```

A codificação dos tratamentos no banco de dados foi da seguinte forma:

Tratamento	Codificação
3DOK1	1
3DOK5	2
3DOK4	3
3DOK7	4
3DOK13	5
COMPOS	6

**Realizando Realizamos** a análise de variância para obter o quadrado médio e o grau de liberdade do resíduo, tem-se:

```
anava <- aov(y~trt)
```

Os argumentos `y` e `trt` da função `MCPtest` têm três tipos de entrada:

- `y` como objeto da função `aov`. Assim, para o argumento `trt` basta **informar informarmos** o nome da coluna que armazena os níveis do tratamento:

```
MCPtest(y = anava, trt = "trt")
```

```

MCP's based on distributions of the studentized midrange
and range
Study:  aov(y ~ trt) ~ "trt"
Study:  aov(y ~ trt) ~ "trt"
Summary:
Means  std r  Min  Max
1 28.82 5.80 5 19.4 33.0
2 23.98 3.78 5 17.7 27.9
3 14.64 4.12 5  9.1 19.4
4 19.92 1.13 5 18.6 21.0
5 13.26 1.43 5 11.6 14.4
6 18.70 1.60 5 16.9 20.8
SNK Midrange Test
Statistics:
Exp.Mean    CV    MSError Df n Stud.Mid    DMS
comp1  19.8867 17.2651 11.7887 24 6    1.0049 1.9862
comp2  19.8867 17.2651 11.7887 24 5    1.0555 2.0640
comp3  19.8867 17.2651 11.7887 24 4    1.1278 2.1750
comp4  19.8867 17.2651 11.7887 24 3    1.2422 2.3506
comp5  19.8867 17.2651 11.7887 24 2    1.4594 2.6841
Groups:
Means      Groups
1 28.82    g1
2 23.98    g1g2
4 19.92    g2
6 18.70    g2g3
3 14.64    g2g3
5 13.26    g3

```

Usando agora a interface **utilizando utilizamos** os mesmos dados: o modelo( $y \sim \text{trt}$ ), tratamento( $\text{trt}$ ), o teste (SNKM) e a opção Modelo, segue na Figura 1.

#### 4.1.2 Opção Variável Resposta

Com a opção 'Variável Resposta', usamos o argumento  $y$  como vetor da variável resposta e o argumento  $\text{trt}$  como vetor dos tratamentos. Devemos informar também na função `MCPtest` o valor do quadrado médio e o grau de liberdade do resíduo nos argumentos `merror` e `dferror`, respectivamente. Assim, temos:

```

anava <- aov(y~trt)
glerror <- df.residual(anava)
qmerror <- deviance(anava) / glerror
# Aplicando os testes
MCPtest(y = y,
trt = trt,
dferror = glerror,
mserror = qmerror)

```

O resultado é idêntico ao apresentado para a opção 'Modelo'.

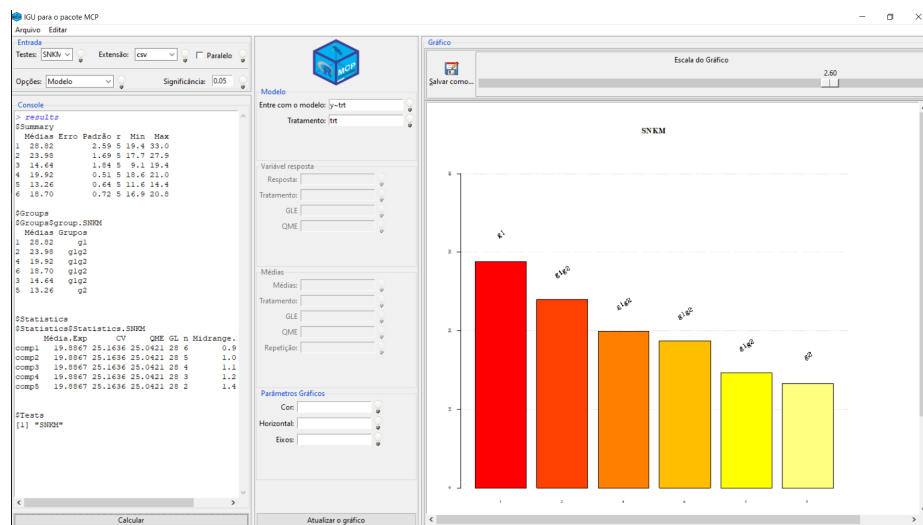


Figura 1 - Teste SNKM feito no Pacote MCPtests utilizando a opção Modelo

**Trabalhando** **Trabalhamos** os dados na interface utilizando os mesmos dados  $y$  como vetor da variável resposta e o argumento  $trt$  como vetor dos tratamentos, o valor do quadrado médio e o grau de liberdade do resíduo nos argumentos  $mserror$  e  $dferror$ , respectivamente, teste(SNKM) e a opção Variável resposta, segue na Figura 2.

### 4.1.3 Opção Média

O  $y$  como vetor de médias dos tratamentos e o argumento  $trt$  como vetor dos tratamentos. Assim, **deve-se informar** **informamos** na função MCPtest que o argumento  $y$  é um vetor de médias, isto é,  $ismean = TRUE$ . **Deve-se também informar** **Inserimos também** o número de repetições, o valor do quadrado médio e o grau de liberdade do resíduo nos argumentos  $replication$ ,  $mserror$  e  $dferror$ , respectivamente. Suponha que no exemplo 3.1, as informações sobre o experimento fossem apenas: a análise de variância, as médias e o número de repetições dos tratamentos, isto é:

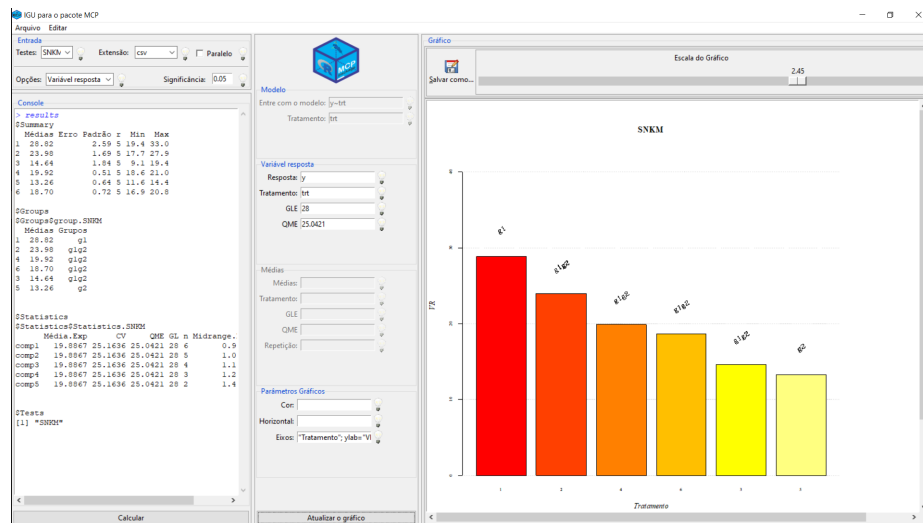


Figura 2 - Teste SNKM feito no Pacote MCPtests utilizando a opção Variável Resposta

anova (anova)

Analysis of Variance Table

Response: y

Df	Sum Sq	Mean Sq	F value	Pr(>F)
trt	5 847.05	169.409	14.37	1.485e-06 ***
Residuals	24 282.93	11.789		

```
# Média dos tratamentos
mean.trt <- c(28.82, 23.98, 14.64, 19.92, 13.26, 18.70)
# Tratamentos
trt <- as.factor(1:6)
# Núm. de repetições
repet <- 5
# Grau de liberdade do resíduo (obtido da ANAVA)
dferror <- 24
# Quadrado médio do resíduo (obtido da ANAVA)
mserror <- 11.78867
```

Com essas informações a função MCPtest também realizará os testes quando ismean = TRUE. Observe os resultados abaixo.

```
# Analise da funcao MCPtest - usando as medias
MCPtest(y = mean.trt,
trt = trt,
```



```
dferror,
mserror,
replication = repet,
ismean = TRUE)
```

Mais uma vez, o resultado será idêntico as opções anteriores. **Trabalhando** **Trabalhamos** os dados na interface utilizando os mesmos dados  $\gamma$  vetor de médias, número de repetições, o valor do quadrado médio e o grau de liberdade do resíduo nos argumentos `replication`, `mserror` e `dferror`, respectivamente. Temos os seguintes resultados:

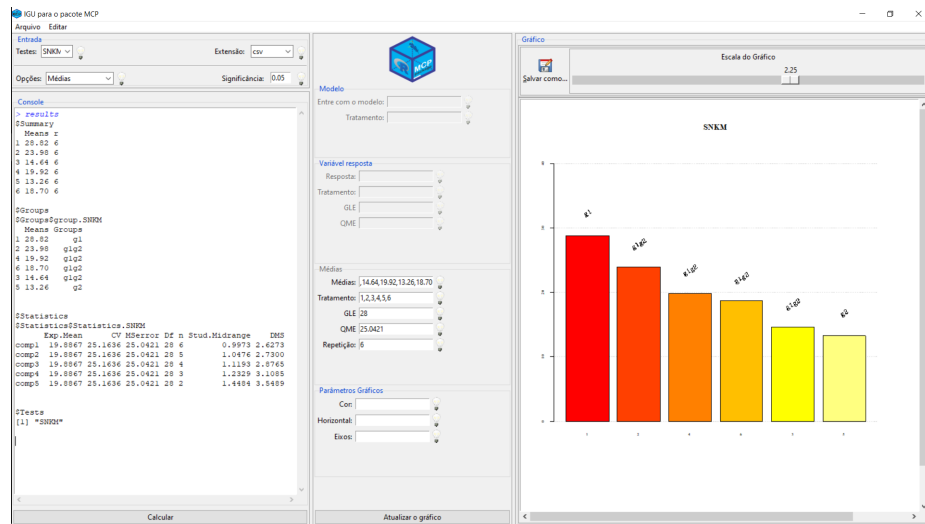


Figura 3 - Teste SNKM feito no Pacote MCPtests utilizando a opção Médias

## 5 CONCLUSÃO

Destacamos que no relatório não dava para apresentar toda a revisão de literatura que fizemos dos procedimentos de comparações múltiplas. Porém, esse projeto será fruto de um desenvolvimento de artigo de revisão e avaliação de desempenho de procedimentos de comparações múltiplas que será enviado para a Revista de Ciência e Agrotecnologia. Mesmo após a finalização do projeto, o trabalho será continuado.

O resultado desse trabalho é o pacote `MCPtests`. Vale lembrar que, o nome do pacote passou por uma mudança, pois na primeira tentativa de submissão ao CRAN, já havia uma pacote com esse nome. Contudo, o pacote sem essa finalidade, estava desativado. Mesmo com isso, é política do R não aceitar submissões de pacotes com o mesmo nome. O número de testes atualmente é 14. Porém, muito outros serão implementados e adicionados no pacote. Devido ao tempo, ainda estamos em fase de implementação da disponibilização da avaliação de desempenho dos testes, uma vez que precisamos desenvolver a rotina interna do pacote para a simulação dos testes.

Para nós, o projeto foi muito enriquecedor, por passarmos por muito desafios estatísticos e computacionais. Isso faz com que ganhemos maturidade em nossas vidas profissionais.

Estamos ainda em fase de submissão ao CRAN, porém o pacote já está disponível em [LINK](#).

## REFERÊNCIAS

- BATISTA, B. D. de O. **Teste de comparações múltiplas baseados na distribuição da *midrange* estudentizada externamente**. 194 p. Tese (Doutorado) — Universidade Federal de Lavras, Lavras, MG, 2016.
- BATISTA, B. D. de O.; FERREIRA, D. F. **midrangeMCP**: Multiple comparisons procedures based on studentized midrange and range distributions. Vienna, Austria, 2020. R package version 3.1. Disponível em: <<http://CRAN.R-project.org/package=midrangeMCP>>.
- BENJAMINI, Y.; BRETZ, F.; SARKAR, S. (Ed.). **Recent Developments in Multiple Comparison Procedures**. Beachwood, Ohio, USA: Institute of Mathematical Statistics, 2004. 156 p.
- BRETZ, F.; HOTHORN, T.; WESTFALL, P. **Multiple comparisons using R**. New York: CRC Press, 2011. 182 p.
- DICKHAUS, T. **Simultaneous Statistical Inference**: with applications in the life sciences. Berlin: Springer, 2014. 180 p.
- DUNCAN, D. B. Multiple range and multiple F tests. **Biometrics**, v. 11, p. 1–42, 1955.
- FERREIRA, E. B.; CAVALCANTI, P. P.; NOGUEIRA, D. A. **ExpDes.pt: Pacote Experimental Designs (Portuguese)**. [S.l.], 2018. R package version 1.2.0. Disponível em: <<https://CRAN.R-project.org/package=ExpDes.pt>>.
- GELMAN, A.; HILL, J.; YAJIMA, M. Why we (usually) don't have to worry about multiple comparisons. **Journal of Research on Educational Effectiveness**, v. 5, p. 189–211, 2012.
- HOCHBERG, Y.; TAMHANE, A. **Multiple comparison procedures**. New York: Wiley, 1987. 450 p.
- HOTHORN, T.; BRETZ, F.; WESTFALL, P. Simultaneous inference in general parametric models. **Biometrical Journal: Journal of Mathematical Methods in Biosciences**, Wiley Online Library, v. 50, n. 3, p. 346–363, 2008.
- HSU, J. C. **Multiple comparisons**: Theory and methods. London: Chapman & Hall/CRC, 1996. 277 p.
- KEULS, M. The use of the “studentized range” in connection with an analysis of variance. **Euphytica**, v. 1, p. 112–122, 1952.

MENDIBURU, F. de; MENDIBURU, M. F. de. Package 'agricolae'. **R package version**, p. 1–2, 2020.

MILLER, R. **Simultaneous statistical inference**. 2. ed. New York: Springer-Verlag, 1981. 299 p.

NEWMAN, D. The distribution of range in samples from a normal population, expressed in terms of an independent estimate of standard deviation. **Biometrika**, v. 31, n. 1/2, p. 20–30, 1939.

O'NEILL, R.; WETHERILL, G. B. The present state of multiple comparison methods. **Journal of the Royal Statistical Society**, v. 33, n. 2, p. 218–250, 1971.

R CORE TEAM. **R**: A language and environment for statistical computing. Vienna, Austria, 2020. Disponível em: <<https://www.R-project.org/>>.

STEEL, R.; TORRIE, J. **Principles and Procedures of Statistics**: A biometrical approach. 2. ed. New York: McGraw-Hill, 1980. 666 p. (McGraw-Hill series in probability and statistics).

STUDENT. Errors in routine analysis. **Biometrika**, v. 19, p. 151–164, 1927.

TUKEY, J. W. The problem of multiple comparisons. Unpublished memorandum in private circulation. 1953.