



UNIVERSIDADE DE SÃO PAULO
Faculdade de Zootecnia e Engenharia de Alimentos
Departamento de Ciências Básicas



ESTATÍSTICA EXPERIMENTAL NO SOFTWARE R

utilizando a interface RStudio

Helena Ribeiro Teixeira

Prof^a. Dr^a. Giovana Fumes Gbantous

Pirassununga - SP

2024

SUMÁRIO

1. SOFTWARE R	3
1.1. INSTALAÇÃO R E RSTUDIO	3
2. R PACOTE BASE	4
2.1. AOV	4
2.2. ANOVA	4
2.3. LM	5
2.4. SUMMARY	6
2.5. STR	6
2.6. INTERACTION PLOT	7
2.7. PLOT	8
2.8. POINTS	9
2.9. CURVE	9
2.10. COEF	10
2.11. RM	10
2.12. TAPPLY	11
3. EXPDES.PT	11
3.1. DIC	12
3.2. DBC	13
3.3. DQL	14
3.4. FATORIAL DUPLO EM DIC	15
3.5. FATORIAL DUPLO EM DBC	15
4. AGRICOLAE	16
4.1. HSD.TEST	17
5. GMODELS	17
5.1. FIT.CONTRAST	18
5.2. MAKE.CONTRASTS	18
6. MULTCOMP	19
6.1. GLHT.TEST	19
7. TIDYVERSE	19
7.1. GGPLOT2	20
7.2. DPLYR	20
8. AGROR	20
8.1. DUNNETT	21
REFERÊNCIAS BIBLIOGRÁFICAS	22

1. SOFTWARE R

Desenvolvido por Ross Ihaka e Robert Gentleman em 1995 na Universidade de Auckland, na Nova Zelândia, o R é um *software* estatístico que proporciona um ambiente interativo para fazer análise de dados. Ele é composto por um sistema básico, integrado pelos pacotes básicos da linguagem, além de outros pacotes complementares. Disponibilizado como *software* livre sob os termos da Licença Pública Geral do GNU da *Free Software Foundation*, o R é distribuído na forma de código fonte, oferecendo um conjunto de instruções escritas de maneira ordenada em uma linguagem de programação lógica (SILVA, 2009).

Ao longo do tempo, o R tem experimentado um desenvolvimento enorme, ampliando suas funcionalidades por meio de uma extensa coleção de pacotes. Esses pacotes possibilitam uma ampla gama de operações, incluindo a manipulação eficaz e fácil de dados, armazenamento, cálculos complexos, análise estatística, e visualização gráfica dos dados. O programa utiliza uma linguagem de programação conhecida como "S", que oferece recursos avançados, incluindo condições, loops, funções recursivas definidas pelo usuário e operações de entrada e saída. Essa linguagem é reconhecida por sua simplicidade e eficácia, tornando o processo de análise de dados mais acessível e intuitivo para os usuários (SILVA, 2009).

O RStudio é um Ambiente de Desenvolvimento Integrado, do inglês - IDE - *Integrated Development Environment* - de código aberto dedicada à linguagem de programação R. Essa plataforma gratuita é projetada por programadores e oferece uma interface simples e organizada, com diversas melhorias, facilitando o uso das ferramentas disponíveis no R.

A interface do RStudio é composta pelo Editor de códigos, no qual scripts são escritos, que apresentam os comandos a serem executados, pelo *Console*, no qual os resultados dos *scripts* são mostrados, além do *Environment*, no qual tem-se a visualização de todo objeto criado no R, e uma aba para gráficos (*Plot*), ajudas (*Help*) entre outros.

1.1. INSTALAÇÃO R E RSTUDIO

A instalação do software R e da interface RStudio pode ser feita pelo site <https://posit.co/download/rstudio-desktop/> (Acesso em 25 de abril de 2024).

2. R PACOTE BASE

O *software* R já vem com pacotes básicos para realizar algumas funções mais simples, entre eles temos o *stats*, utilizado para fazer algumas análises estatísticas através das funções disponibilizadas, e o *lattice* é utilizado para criar gráficos complexos como gráficos de superfície, como gráficos de dispersão 3D e gráficos de superfície 2D, e de fácil personalização, como mudança de cores e fontes (BECKERS; CHAMBERS; WILKS, 1988; SARKAR, 2008). A seguir algumas funções com seus argumentos foram explicitadas.

2.1. AOV

Esta função é utilizada para análise de variância por meio de ajuste de modelos lineares. Esta função é para modelos balanceados, e pode ser difícil interpretar sem balanceamento, portanto, não é recomendado o uso com parcelas faltantes (CHAMBERS; FREENY; HEIBERGER, 1992). A função é definida por:

$$aov(formula, data = \text{NULL}, projections = \text{FALSE}, \\ qr = \text{TRUE}, contrasts = \text{NULL}, \dots)$$

em que:

- **formula:** é a fórmula especificando o modelo;
- **data:** banco de dados no qual as variáveis especificadas na fórmula serão encontradas. Se estiver faltando, as variáveis são procuradas da maneira padrão;
- **projections:** devem ser retornadas as projeções, quando “TRUE” serão retornadas as projeções. default = FALSE;
- **qr:** Deve ser retornado à decomposição QR. Quando “TRUE” será retornada à decomposição. default = TRUE;
- **contrasts:** é uma lista de contrastes que serão usados para alguns dos fatores na fórmula.

2.2. ANOVA

Este comando é usado para explicitar tabelas de análise de variância para um ou mais objetos de modelo ajustado. Ele retorna um objeto contendo tabelas de análise de variância. Quando dado um único argumento, ele produz uma tabela que testa se os termos do modelo são significantes. Quando dado uma sequência de objetos, anova testa os modelos um contra o outro na ordem especificada. A comparação entre dois ou mais modelos será válida apenas se eles

forem ajustados ao mesmo conjunto de dados (CHAMBERS; HASTIE, 1992). A função é definida por:

anova(object, ...)

em que:

- **object:** é um objeto contendo os resultados retornados por uma função de ajuste de modelo (por exemplo, `lm` ou `glm`);
- **...** : objetos adicionais do mesmo tipo.

2.3. LM

Esta função é utilizada para ajustar modelos lineares, incluindo multivariados. Pode ser usado para realizar regressão, análise de variância de estrato único e análise de covariância (CHAMBERS; HASTIE, 1992). A função é definida por:

*lm(formula, data, subset, weights, na.action, method = "qr", model = **TRUE**, x = **FALSE**, y = **FALSE**, qr = **TRUE**, singular.ok = **TRUE**, contrasts = **NULL**, offset, ...)*

em que:

- **formula:** objeto da classe "formula" (ou que pode ser convertido para essa classe): uma descrição simbólica do modelo a ser ajustado;
- **data:** Um *data frame*, lista ou ambiente opcional (ou objeto coercível por *as.data.frame* para um *data frame*) contendo as variáveis no modelo. Se não encontrado em *data*, as variáveis são extraídas de *environment* (formula), tipicamente o ambiente de onde `lm` é chamado;
- **subset:** Um vetor opcional especificando um subconjunto de observações a ser usado no processo de ajuste;
- **weights:** Um vetor opcional de pesos a ser usado no processo de ajuste. Deve ser `NULL` ou um vetor numérico. Se não for `NULL`, mínimos quadrados ponderados é usado com pesos *weights* (isto é, minimizando $\sum(w \cdot e^2)$); caso contrário, mínimos quadrados ordinários é usado;
- **na.action:** Uma função que indica o que deve acontecer quando os dados contêm NAs. O padrão é definido pela configuração *na.action* de *options*, e é *na.fail* se não estiver configurado. O padrão é *na.omit*. Outro valor possível é `NULL`, nenhuma ação. O valor *na.exclude* pode ser útil;

- **method:** O método a ser usado; para ajuste, atualmente apenas *method = "qr"* é suportado; *method = "model.frame"* retorna o quadro do modelo (o mesmo que com *model = TRUE*);
- **model, x, y, qr:** Lógicos. Se *TRUE*, os componentes correspondentes do ajuste (o quadro do modelo, a matriz do modelo, a resposta, a decomposição QR) são retornados;
- **singular.ok:** Lógico. Se *FALSE* (o padrão em S, mas não em R), um ajuste singular é um erro;
- **contrasts:** Uma lista opcional;
- **offset:** Pode ser usado para especificar um componente a priori conhecido a ser incluído no preditor linear durante o ajuste. Deve ser *NULL* ou um vetor numérico ou matriz de extensões correspondentes às da resposta. Um ou mais termos *offset* podem ser incluídos na fórmula em vez disso ou também, e se mais de um for especificado, sua soma é usada;

2.4. SUMMARY

Este comando é utilizado para produzir sumários dos resultados do modelo utilizado (CHAMBERS; HASTIE, 1992), definido por:

summary(object, ...)

em que:

- **object:** é um objeto contendo os resultados retornados por uma função de ajuste de modelo (por exemplo, *lm* ou *glm*);
- **...** : objetos adicionais do mesmo tipo.

2.5. STR

Este comando é utilizado para exibir compactamente a estrutura interna de um objeto R, uma função de diagnóstico e uma alternativa ao *summary* (e em certa medida, *dput*). Idealmente, apenas uma linha para cada estrutura 'básica' é exibida. É especialmente adequado para exibir de forma compacta o conteúdo (abreviado) de listas (possivelmente aninhadas). Fornece uma saída razoável para qualquer objeto R (MAECHLER, 1990). A função é definida por:

str(object, ...)

em que:

- **object:** Qualquer objeto R sobre o qual deseja obter algumas informações;

- ...: Outros argumentos potenciais (necessários por razões de Método/Genérico).

2.6. INTERACTION PLOT

Este comando é utilizado para plotar a média (ou outro resumo) da resposta para combinações bidirecionais de fatores, ilustrando assim possíveis interações (CHAMBERS; FREENY; HEIBERGER, 1992), sendo definido por:

```
interaction.plot(x.factor, trace.factor, response, fun = mean, type = c("l", "p", "b", "o", "c"),
  legend = TRUE, trace.label = deparse1(substitute(trace.factor)), fixed = FALSE, xlab =
  deparse1(substitute(x.factor)), ylab = ylabel, ylim = range(cells, na.rm = TRUE), lty = nc:1,
  col = 1, pch = c(1:9, 0, letters), xpd = NULL, leg.bg = par("bg"), leg.bty = "n", xtick =
  FALSE, xaxt = par("xaxt"), axes = TRUE, ...)
```

em que:

- **x.factor:** fator cujos níveis formarão o eixo x;
- **trace.factor:** fator cujos níveis formarão as linhas de traço;
- **response:** variável numérica que fornece a resposta;
- **fun:** função para calcular o resumo. Deve retornar um único valor real;
- **type:** tipo de gráfico (ver *plot.default*): linhas ou pontos ou ambos;
- **legend:** Lógico. Deve incluir uma legenda.
- **trace.label:** Rótulo geral para a legenda;
- **fixed:** Lógico. A legenda deve estar na ordem dos níveis de trace.factor (TRUE) ou na ordem das linhas de traço em suas extremidades direitas (FALSE, o padrão)
- **xlab, ylab:** rótulos dos eixos x e y do gráfico, cada um com um padrão sensato;
- **ylim:** Numérico de comprimento 2, dando os limites y para o gráfico;
- **lty:** Tipo de linha para as linhas desenhadas, com um padrão sensato;
- **col:** A cor a ser usada para o gráfico;
- **pch:** vetor de símbolos ou caracteres de plotagem, com um padrão sensato;
- **xpd:** Determina o comportamento de recorte para a legenda usada, ver *par(xpd)*. Por padrão, a legenda não é cortada na borda da figura;
- **leg.bg, leg.bty:** Argumentos passados para a função *legend()*;
- **xtick:** Lógico. Devem ser usadas marcas de escala no eixo x
- **xaxt, axes, ...:** Parâmetros gráficos a serem passados para as rotinas de plotagem.

2.7. PLOT

Este comando é utilizado para plotagem de objetos R. Para gráficos de dispersão simples, *plot.default* será usado. No entanto, existem métodos de plotagem para muitos objetos R, incluindo funções, *data.frames*, objetos de densidade, etc. A maioria desses métodos é implementada usando gráficos tradicionais (o pacote *graphics*), mas não é obrigatório, sendo definido por:

$$\text{plot}(x, y, \dots)$$

em que:

- **x**: coordenadas dos pontos no gráfico. Alternativamente, pode ser fornecida uma estrutura de plotagem única, função ou qualquer objeto R com um método de plotagem;
- **y**: coordenadas y dos pontos no gráfico, opcional se x for uma estrutura apropriada;
- **...**: Argumentos a serem passados para métodos, como parâmetros gráficos (ver par).

Muitos métodos aceitarão os seguintes argumentos:

- **main**: título geral para o gráfico;
- **sub**: subtítulo para o gráfico;
- **xlab**: título para o eixo x;
- **ylab**: título para o eixo y;
- **asp**: razão de aspecto y/x;
- **type**: O tipo de gráfico que deve ser desenhado. Tipos possíveis são:
 - "p" para pontos;
 - "l" para linhas;
 - "b" para ambos;
 - "c" para a parte de linhas sozinha de "b";
 - "o" para ambos sobrepostos;
 - "h" para linhas verticais estilo histograma (ou alta densidade);
 - "s" para degraus;
 - "S" para outros degraus;
 - "n" para nenhuma plotagem.

Todos os outros tipos dão um aviso ou um erro; usar, por exemplo, *type = "punkte"* é equivalente a *type = "p"* para compatibilidade com S. Note que alguns métodos, por exemplo, *plot.factor*, não aceitam isso.

2.8. POINTS

Esta função é utilizada para desenhar uma sequência de pontos nas coordenadas especificadas. Os caracteres especificados são plotados, centralizados nas coordenadas. A função é definida por:

points(x, ...)

em que:

- **x, y:** Vetores de coordenadas dos pontos a serem plotados;
- **type:** Caractere indicando o tipo de plotagem; na verdade, qualquer um dos tipos conforme em *plot.default*;
- **...:** Outros parâmetros gráficos podem ser fornecidos como argumentos.

As coordenadas podem ser passadas em uma estrutura de plotagem (uma lista com componentes x e y), uma matriz de duas colunas, uma série temporal, etc.

2.9. CURVE

Esta função é utilizada para desenhar uma curva correspondente a uma função no intervalo *[from, to]*, *curve* também pode plotar uma expressão na variável *xname*, padrão 'x'. A função é definida por:

curve(expr, from = NULL, to = NULL, n = 101, add = FALSE, type = "l", xname = "x", xlab = xname, ylab = NULL, log = NULL, xlim = NULL, ...)

em que:

- **expr:** nome de uma função, ou uma chamada ou expressão escrita como uma função de x que será avaliada para um objeto do mesmo comprimento que x;
- **x:** Uma função numérica de R vetorizada;
- **y:** Alias para from para compatibilidade com plot;
- **from, to:** O intervalo sobre o qual a função será plotada;
- **n:** Inteiro; o número de valores de x nos quais avaliar;
- **add:** Lógico; se *TRUE*, adiciona a um gráfico já existente; se NA, inicia um novo gráfico tomando os padrões para os limites e escala logarítmica do eixo x do gráfico anterior. Considerado como *FALSE* (com um aviso se um valor diferente for fornecido) se nenhum dispositivo gráfico estiver aberto;

- **xlim:** NULL ou um vetor numérico de comprimento 2; se não for NULL, fornece os padrões para c(from, to) e, a menos que add = TRUE, seleciona os limites x do gráfico;
- **type:** Tipo de gráfico;
- **xname:** Cadeia de caracteres dando o nome a ser usado para o eixo x;
- **xlab, ylab, log, ...:** Rótulos e parâmetros gráficos também podem ser especificados como argumentos.

2.10. COEF

Esta é utilizada para extrair coeficientes de modelo de objetos retornados por funções de modelagem. `coefficients` é um alias para ela. A função é definida por:

coef(object, ...)

em que:

- **object:** Um objeto para o qual a extração de coeficientes do modelo faz sentido;
- **...:** Outros argumentos.

2.11. RM

Utilizada para remover objetos. Estes podem ser especificados sucessivamente como *strings* de caracteres, ou em uma lista de vetores de caracteres, ou através de uma combinação de ambos. Todos os objetos assim especificados serão removidos. `remove` é um alias para ela. A função é definida por:

rm(..., list = character(), pos = -1, envir = as.environment(pos), inherits = FALSE)

em que:

- **...:** os objetos a serem removidos, como nomes (não citados) ou *strings* de caracteres (citados);
- **list:** um vetor de caracteres (ou NULL) nomeando os objetos a serem removidos;
- **pos:** onde fazer a remoção. Por padrão, usa o ambiente atual;
- **envir:** o ambiente a ser usado;
- **inherits:** os frames de fechamento do ambiente devem ser inspecionados

2.12. TAPPLY

Esta é utilizada para aplicar uma função, geralmente para calcular uma estatística única, como média, mediana ou desvio padrão, dentro dos níveis de um fator ou dentro das combinações de níveis de dois ou mais fatores para produzir uma tabela de estatísticas. Esta função fornece uma interface de fórmula para a função padrão `tapply` do R (FOX, 2023). A função é definida por:

$$\text{tapply}(\text{formula}, \text{fun}, \text{data}, \text{na.action} = \text{na.pass}, \dots, \text{targs} = \text{list}())$$

em que:

- **formula:** uma fórmula de dois lados no formato `variável ~ fator.1 + fator.2 + ... + fator.n` ou, equivalentemente, `variável ~ fator.1*fator.2* ... *fator.n.`;
- **fun:** uma função, como *mean*, *median* ou *sd*, que recebe um vetor como primeiro argumento e tipicamente retorna um único número como seu valor;
- **data:** um *data frame* opcional no qual encontrar a variável e o(s) fator(es);
- **na.action:** uma função para lidar com valores ausentes, como em funções de modelagem estatística como *lm*; o padrão é `na.pass`;
- **...:** argumentos para passar para a função fornecida no argumento `fun`, como (comumente) `na.rm=TRUE`;
- **targs:** uma lista de argumentos opcionais para passar para `tapply`.

A função especificada por `fun` é aplicada aos valores da variável do lado esquerdo na fórmula dentro dos níveis (combinação de) dos fatores fornecidos no lado direito da fórmula, produzindo uma tabela de estatísticas.

3. EXPDES.PT

O pacote *ExpDes.pt* oferece uma ferramenta abrangente para análise de experimentos balanceados sob modelos fixos, e diversos tipos de delineamentos experimentais, incluindo DIC, DBC, DQL, esquema fatorial duplo, parcelas subdivididas e esquema fatorial triplo, tanto com, como sem tratamentos adicionais. Para cada delineamento, realiza-se análise de variância e comparação de médias, ajustando modelos de regressão até o terceiro grau para tratamentos quantitativos ou conduzindo testes de comparação múltipla, como o teste de Tukey, para tratamentos qualitativos (NOGUEIRA; FERREIRA; CAVALCANTI, 2021).

O pacote ExpDes, realiza as mesmas funções do ExpDes.pt, sendo a única diferença no idioma dos resultados, sendo em inglês no primeiro e em português no segundo (NOGUEIRA; FERREIRA; CAVALCANTI, 2021).

3.1. DIC

Esta função é utilizada para analisar experimentos em delineamento inteiramente casualizado, balanceado com um só fator, considerando um modelo fixo. A função é definida por:

```
dic(trat, resp, quali = TRUE, mcomp = "tukey", nl = FALSE, hvar = "bartlett",  
sigT = 0.05, sigF = 0.05, unfold = NULL)
```

em que:

- **trat:** é o vetor contendo os tratamentos;
- **resp:** é o vetor com os dados (variável resposta);
- **quali:** quando "TRUE", os tratamentos são entendidos como qualitativos, e quando "FALSE", quantitativos. default = "TRUE";
- **mcomp:** permite escolher o teste de comparação múltipla. default = "tukey". Outras opções: teste LSD ("lsd"), teste LSDB ("lsdb"), teste de Duncan ("duncan"), teste de SNK ("snk"), teste de Scott-Knott ("sk"), teste de comparações múltiplas bootstrap ("ccboot") e o teste de Calinski e Corsten baseado na distribuição F ("ccf");
- **nl:** Escolhe se modelos não lineares devem ser ajustados aos fatores quantitativos. O default = FALSE;
- **hvar:** permite escolher o teste de homogeneidade de variâncias. default = teste de O'Neill e Mathews ("oneillmathews");
- **SigT:** é a significância a ser adotada pelo teste de comparação múltipla de médias. default = 0,05;
- **SigF:** é a significância a ser adotada pelo teste F da ANOVA. default = 0,05;
- **Unfold:** Orienta os desdobramentos após a análise de variância. Se NULL (default), são feitas as análises recomendadas; se '0', e feita apenas a análise de variância; se '1', os efeitos simples são estudados.

Na saída, a função retorna os valores da análise de variância do DIC, o teste de normalidade de Shapiro-Wilk para os resíduos do modelo, o ajuste de modelos de regressão (caso de tratamentos quantitativos) ou os testes de comparação de médias (caso de tratamentos

qualitativos): teste de Tukey, teste de Duncan, teste t de Student (LSD), teste t de Bonferroni, teste de Student-Newman-Keuls(SNK), teste de Scott-Knott e teste de comparações múltiplas bootstrap (NOGUEIRA; FERREIRA; CAVALCANTI, 2021).

3.2. DBC

Esta é utilizada para analisar experimentos em Delineamento em Blocos Casualizados balanceado com um só fator, considerando o modelo fixo (NOGUEIRA; FERREIRA; CAVALCANTI, 2021). A função é definida por:

```
dbc( trat, bloco, resp, quali = TRUE, mcomp = "tukey", nl = FALSE, hvar =  
    "oneillmathews", sigT = 0.05, sigF = 0.05, unfold = NULL)
```

em que:

- **trat:** é o vetor contendo os tratamentos;
- **bloco:** é o vetor contendo os blocos;
- **resp:** é o vetor com os dados (variável resposta);
- **quali:** quando "TRUE", os tratamentos são entendidos como qualitativos, e quando "FALSE", quantitativos. default = "TRUE";
- **mcomp:** permite escolher o teste de comparação múltipla. default = "tukey". Outras opções: teste LSD ("lsd"), teste LSDB ("lsdb"), teste de Duncan ("duncan"), teste de SNK ("snk"), teste de Scott-Knott ("sk"), teste de comparações múltiplas bootstrap ("ccboot") e o teste de Calinski e Corsten baseado na distribuição F ("ccf");
- **nl:** Escolhe se modelos não lineares devem ser ajustados aos fatores quantitativos. O default = FALSE;
- **hvar:** permite escolher o teste de homogeneidade de variâncias. default = teste de O'Neill e Mathews ("oneillmathews");
- **SigT:** é a significância a ser adotada pelo teste de comparação múltipla de médias. default = 0,05;
- **SigF:** é a significância a ser adotada pelo teste F da ANOVA. default = 0,05;
- **Unfold:** Orienta os desdobramentos após a análise de variância. Se NULL (default), são feitas as análises recomendadas; se '0', e feita apenas a análise de variância; se '1', os efeitos simples são estudados.

A função retorna os valores da análise de variância do DBC, e outros resultados explicitados no **item 3.1**.

3.3. DQL

Esta é utilizada para analisar experimentos em Delineamento em Quadrado Latino balanceado com um só fator, considerando o modelo fixo (NOGUEIRA; FERREIRA; CAVALCANTI, 2021). A função é definida por:

*dql(trat, linha, coluna, resp, quali = TRUE, mcomp = "tukey", sigT = 0.05,
sigF = 0.05, unfold = NULL)*

em que:

- **trat:** é o vetor contendo os tratamentos;
- **linha:** é o vetor contendo as linhas;
- **coluna:** é o vetor contendo as colunas;
- **resp:** é o vetor com os dados (variável resposta);
- **quali:** quando "TRUE", os tratamentos são entendidos como qualitativos, e quando "FALSE", quantitativos. default = "TRUE";
- **mcomp:** permite escolher o teste de comparação múltipla. default = "tukey". Outras opções: teste LSD ("lsd"), teste LSDb ("lsdb"), teste de Duncan ("duncan"), teste de SNK ("snk"), teste de Scott-Knott ("sk"), teste de comparações múltiplas bootstrap ("ccboot") e o teste de Calinski e Corsten baseado na distribuição F ("ccf");
- **nl:** Escolhe se modelos não lineares devem ser ajustados aos fatores quantitativos. O default = FALSE;
- **hvar:** permite escolher o teste de homogeneidade de variâncias. default = teste de O'Neill e Mathews ("oneillmathews");
- **SigT:** é a significância a ser adotada pelo teste de comparação múltipla de médias. default = 0,05;
- **SigF:** é a significância a ser adotada pelo teste F da ANOVA. default = 0,05;
- **Unfold:** Orienta os desdobramentos após a análise de variância. Se NULL (default), são feitas as análises recomendadas; se '0', e feita apenas a análise de variância; se '1', os efeitos simples são estudados.

A função retorna os valores da análise de variância do DQL, e outros resultados explicitados no **item 3.1**.

3.4. FATORIAL DUPLO EM DIC

Esta é utilizada para analisar experimentos em fatorial duplo em Delineamento Inteiramente Casualizado balanceado, considerando o modelo fixo (NOGUEIRA; FERREIRA; CAVALCANTI, 2021). A função é definida por:

```
fat2.dic( fator1, fator2, resp, quali = c(TRUE, TRUE), mcomp = "tukey",  
          fac.names = c("F1", "F2"), sigT = 0.05, sigF = 0.05, unfold = NULL)
```

em que:

- **fator1:** é o vetor numérico ou complexo contendo os níveis do fator 1;
- **fator2:** é o vetor numérico ou complexo contendo os níveis do fator 2;
- **resp:** é o vetor numérico ou complexo contendo a variável resposta;
- **quali:** se TRUE (default) na primeira posição, os níveis do fator 1 são entendidos como qualitativos, se FALSE, quantitativos; da mesma forma, a segunda posição é referente aos níveis do fator 2;
- **mcomp:** Permite escolher o teste de comparação múltipla; o default é o teste de Tukey, contudo tem-se como outras opções: o teste LSD ('lsd'), o teste LSDB ('lsdb'), o teste de Duncan ('duncan'), o teste de SNK ('snk'), o teste de Scott-Knott ('sk'), o teste de comparações múltiplas bootstrap ('ccboot') e o teste de Calinski e Corsten baseado na distribuição F ('ccf');
- **fac.names:** Permite nomear os fatores 1 e 2;
- **sigT:** Significância a ser adotada pelo teste de comparação múltipla de médias; o default é 5%;
- **sigF:** Significância a ser adotada pelo teste F da ANOVA; o default é 5%;
- **unfold:** Orienta os desdobramentos após a análise de variância. Se NULL (default), são feitas as análises recomendadas; se '0', é feita apenas a análise de variância; se '1', os efeitos simples são estudados; se '2', a interação dupla é estudada.

A função retorna os valores da análise de variância do DIC em questão, e outros resultados explicitados no **item 3.1**.

3.5. FATORIAL DUPLO EM DBC

Esta é utilizada para analisar experimentos em fatorial duplo em Delineamento em Blocos Casualizados balanceado, considerando o modelo fixo (NOGUEIRA; FERREIRA; CAVALCANTI, 2021). A função é definida por:

```
fat2.dbc( fator1, fator2, bloco, resp, quali = c(TRUE, TRUE), mcomp = "tukey",  
fac.names = c("F1", "F2"), sigT = 0.05, sigF = 0.05, unfold = NULL)
```

em que:

- **fator1:** é o vetor numérico ou complexo contendo os níveis do fator 1;
- **fator2:** é o vetor numérico ou complexo contendo os níveis do fator 2;
- **resp:** é o vetor numérico ou complexo contendo a variável resposta;
- **quali:** se TRUE (default) na primeira posição, os níveis do fator 1 são entendidos como qualitativos, se FALSE, quantitativos; da mesma forma, a segunda posição é referente aos níveis do fator 2;
- **mcomp:** Permite escolher o teste de comparação múltipla; o default é o teste de Tukey, contudo tem-se como outras opções: o teste LSD ('lsd'), o teste LSDb ('lsdb'), o teste de Duncan ('duncan'), o teste de SNK ('snk'), o teste de Scott-Knott ('sk'), o teste de comparações múltiplas bootstrap ('ccboot') e o teste de Calinski e Corsten baseado na distribuição F ('ccf');
- **fac.names:** Permite nomear os fatores 1 e 2;
- **sigT:** Significância a ser adotada pelo teste de comparação múltipla de médias; o default é 5%;
- **sigF:** Significância a ser adotada pelo teste F da ANOVA; o default é 5%;
- **unfold:** Orienta os desdobramentos após a análise de variância. Se NULL (default), são feitas as análises recomendadas; se '0', é feita apenas a análise de variância; se '1', os efeitos simples são estudados; se '2', a interação dupla é estudada.

A função retorna os valores da análise de variância do DBC em questão, e outros resultados explicitados no **item 3.1**.

4. AGRICOLAE

O pacote *agricolae* foi criado principalmente para experimentos agrícolas, experimentos de campo em agricultura e melhoramento de plantas, oferece uma ampla gama de funcionalidades tanto para projetos experimentais quanto para análises sensoriais (MENDIBURU, 2023).

Com recursos abrangentes em designs experimentais, ele pode ser utilizado para diferentes tipos de delineamentos, como DBC, DQL, fatorial e blocos incompletos balanceados

(BIB). Além disso, o pacote inclui procedimentos para comparação de tratamentos e realização de diversos testes, incluindo os não paramétricos (MENDIBURU, 2023).

4.1. HSD.TEST

Esta função é utilizada para fazer a comparação entre as médias dos tratamentos pelo teste de Tukey. O nível de significância default = 0,05 (MENDIBURU, 2023). A função é definida por:

```
HSD.test(y, trt, DError, MSerror, alpha = 0.05, group=TRUE, main =  
NULL, unbalanced=FALSE, console=FALSE)
```

em que:

- **y:** É a fórmula especificando o modelo (aov ou lm) ou a resposta da unidade experimental;
- **trt:** Constante (apenas se y=modelo) ou vetor de tratamento aplicado a cada unidade experimental;
- **DError:** Graus de liberdade do erro;
- **MSerror:** Quadrado médio do erro;
- **alpha:** Nível de significância;
- **group:** Formação de grupos de tratamento (TRUE ou FALSE);
- **main:** Título;
- **unbalanced:** Indica se as replicações são desiguais (TRUE ou FALSE);
- **console:** saída

5. GMODELS

O pacote *gmodels* é uma ferramenta abrangente para a realização de análises estatísticas e sumarização de dados. Ele fornece uma série de funções úteis para explorar e entender conjuntos de dados, especialmente através de tabelas de contingência e estatísticas associadas, e inclui funções para calcular testes estatísticos como o qui-quadrado (R CORE TEAM, 2023).

Ele faz a análise robusta das relações entre variáveis e facilita a interpretação dos resultados com opções de visualização e relatórios detalhados (R CORE TEAM, 2023).

5.1. FIT.CONTRAST

Utilizada para calcular o(s) contraste(s) especificado(s) refazendo o ajuste do modelo com os argumentos apropriados. A função é definida por:

fit.contrast(model, varname, coeff, showall, conf.int, df, ...)

em que:

- **model:** Objeto de regressão (lm, glm, aov, lme) para o qual o(s) contraste(s) será(ão) calculado(s);
- **varname:** Nome da variável;
- **coeff:** Vetor ou matriz especificando contrastes (um por linha);
- **showall:** Retornar todos os coeficientes de regressão. Se *TRUE*, todos os coeficientes do modelo serão retornados. Se *FALSE* (o padrão), apenas os coeficientes correspondentes ao contraste especificado serão retornados;
- **conf.int:** Valor numérico entre (0,1) ou NULL. Se um valor numérico for especificado, intervalos de confiança com a probabilidade de cobertura nominal conf.int serão calculados. Se NULL, intervalos de confiança não serão calculados;
- **df:** Booleano indicando se deve retornar uma coluna contendo os graus de liberdade;
- **...:** Argumentos opcionais fornecidos pelos métodos.

5.2. MAKE.CONTRASTS

Esta é utilizada para montar e converter contrastes estatísticos para a linguagem computacional do R. A função é definida por:

make.contrasts(contr, how.many = ncol(contr))

em que:

- **contr:** vetor ou matriz especificando contrastes (um por linha);
- **how.many:** dimensões da matriz de contrastes desejada. Isso deve ser igual ao número de níveis da variável fator alvo.

6. MULTCOMP

O pacote *multcomp* realiza comparações múltiplas e testes de hipóteses lineares gerais em modelos paramétricos, incluindo modelos lineares generalizados, modelos lineares mistos e modelos de sobrevivência. Ele oferece métodos para especificar e testar hipóteses lineares de forma flexível, para comparações gerais e múltiplas (CHRISTENSEN, 2021; FOX; WEISBERG, 2019).

6.1. GLHT.TEST

Esta função é utilizada para hipóteses lineares gerais e comparações múltiplas para modelos paramétricos, incluindo modelos lineares generalizados, modelos de efeitos mistos lineares e modelos de sobrevivência. A função é definida por:

glht(model, linfct, alternative = c("two.sided", "less", "greater"), rhs = 0, ...)

em que:

- **model:** um modelo ajustado, por exemplo, um objeto retornado por `lm`, `glm`, ou `aov` etc. Assume-se que métodos `coef` e `vcov` estejam disponíveis para o modelo;
- **linfct:** uma especificação das hipóteses lineares a serem testadas. Funções lineares podem ser especificadas por meio da matriz de coeficientes ou por descrições simbólicas de uma ou mais hipóteses lineares;
- **alternative:** uma *string* de caractere especificando a hipótese alternativa, deve ser uma das seguintes: *"two.sided"* (padrão), *"greater"* ou *"less"*. Pode especificar apenas a letra inicial;
- **rhs:** um vetor numérico opcional especificando o lado direito da hipótese;
- **...:** argumentos adicionais para a função modelarem em todos os métodos `glht`. Para a função `mcp`, comparações múltiplas são definidas por matrizes ou descrições simbólicas que especificam contrastes de níveis de fatores onde os argumentos correspondem aos nomes dos fatores.

7. TIDYVERSE

O pacote *tidyverse* é uma coleção de pacotes do R projetados para a ciência de dados, desenvolvidos por Hadley Wickham e outros colaboradores. Esses pacotes compartilham uma

base comum de design, gramática e estrutura de dados. O *tidyverse* inclui pacotes populares como *dplyr*, *ggplot2*, *tidyr*, *purrr* e *tibble*, cada um com funções específicas para manipulação, exploração e comunicação de dados (WICKHAM et al., 2019; WICKHAM; GROLEMUND, 2016).

O pacote *dplyr*, por exemplo, fornece funções intuitivas e eficientes para filtrar, selecionar, transformar e resumir dados. Já o *ggplot2* é um pacote poderoso para visualização de dados, criando gráficos sofisticados e personalizados, adicionando camadas de dados, mapeamentos estéticos, transformações estatísticas e objetos geométricos (WICKHAM et al., 2019; WICKHAM; GROLEMUND, 2016).

Além disso, o *tidyverse* enfatiza a importância de manter os dados em um formato "*tidy*", em que cada variável é uma coluna, cada observação é uma linha e cada tipo de unidade observacional é uma tabela. Essa estrutura facilita a compreensão e manipulação dos dados, tornando o processo de análise mais eficiente (WICKHAM et al., 2019; WICKHAM; GROLEMUND, 2016).

7.1. GGLOT2

O *ggplot2* é parte da coleção de pacotes *tidyverse*, amplamente utilizada para análise de dados e *machine learning*. Foi desenvolvido para construção de gráficos em camadas, ou seja, os elementos do gráfico são inseridos em camadas, que se sobrepõem para a construção do gráfico. Por razão desta estrutura e a disposição padronizada das funções, é de fácil personalização e mais intuitivo para o uso (DAMIANI et. al., 2015).

As camadas são formadas por dados, designs estéticos e objetos geométricos, ajustes de posicionamento, entre outras diversas personalizações (GONZAGA et al., 2021).

7.2 DPLYR

O *dplyr* é parte da coleção de pacotes *tidyverse*. Foi criado principalmente para trabalhar com objetos semelhantes a data frames, tanto na memória quanto fora dela, além de incluir procedimentos para manipulação de dados (DAMIANI, 2022; WICKHAM, 2023).

8. AGROR

O pacote *AgroR* é utilizado para realizar a análise de delineamentos experimentais, experimentos em esquemas fatoriais, podendo ser duplo e triplo, experimentos em esquema de

parcelas subdivididas, análise de regressão linear, entre outros. Além disso, o pacote inclui procedimentos para análise de variância e realização de diversos testes, incluindo os não paramétricos. Pode ainda fornecer medidas descritivas e gráficos (SHIMIZU, 2024).

8.1. DUNNETT

Esta função é utilizada para fazer a comparação entre a média do tratamento controle e as médias dos demais tratamentos pelo teste de Dunnett. A função é definida por:

dunnett(trat, resp, control, model = "DIC", block = NA)

em que:

- **trat:** é o vetor contendo os tratamentos;
- **resp:** é o vetor com os dados (variável resposta);
- **control:** Tratamento considerado como controle;
- **model:** é o tipo de delineamento experimental utilizado (DIC, DBC ou DQL);
- **block:** é o vetor contendo os blocos.

REFERÊNCIAS BIBLIOGRÁFICAS

BECKER, R.; CHAMBERS, J.; WILKS, A. **The New S Language**. Wadsworth & Brooks/Cole, 1988.

CHAMBERS, J.; FREENY, A.; HEIBERGER, R. **Analysis of variance: designed experiments**. Wadsworth & Brooks/Cole, 1992.

CHAMBERS, J.; HASTIE, T. **Statistical Models in S**. Wadsworth & Brooks/Cole, 1992.

CHRISTENSEN, R. **Multcomp: Simultaneous Inference in General Parametric Models**. 2021. Disponível em: <https://CRAN.R-project.org/package=multcomp>. Acesso em: 30 jun. 2024.

DAMIANI, A. et al. **Ciência de Dados em R**, 2022. Disponível em: <https://livro.curso-r.com/8-1-o-pacote-ggplot2.html>. Acesso em: 23 abr. 2024.

FARIA, P; PARGA, J. **Introdução à Linguagem R: seus fundamentos e sua prática**. 2. ed. Belo Horizonte: 2021. ISBN 978-65-00-12606-8. Disponível em: https://pedro-faria.netlify.app/pt/publication/book/introducao_linguagem_r/. Acesso em: 13 nov. 2023.

FERREIRA, E.; CAVALCANTI, P.; NOGUEIRA, D. **Pacote Experimental Designs** (Português). [S.l.], 2021. Disponível em: <https://cran.r-project.org/web/packages/ExpDes.pt/ExpDes.pt.pdf>. Acesso em: 04 abr. 2024.

FOX, J.; WEISBERG, S. **An R Companion to Applied Regression**. 3 ed. Sage, 2019.

FOX, J. **CAR**. 2023. Disponível em: <https://cran.r-project.org/web/packages/car/car.pdf>. Acesso em: 30 jun. 2024.

GONZAGA, S. et al. **Ciências de Dados com R**. IBPAD, 2021. Disponível em: <http://sillasgonzaga.com/material/cdr>. Acesso em: 23 abr. 2024.

MAECHLER, M. **STAT MATH**, 1990.

MENDIBURU, F. de. **Statistical Procedures for Agricultural Research**. [S.l.], 2023. Disponível em: <https://cran.r-project.org/web/packages/agricolae/agricolae.pdf>. Acesso em: 04 abr. 2024.

R CORE TEAM. **gmodels: Various R Programming Tools for Model Fitting**. CRAN, 2023. Disponível em: <https://cran.r-project.org/web/packages/gmodels/gmodels.pdf>. Acesso em: 30 jun. 2024.

SARKAR, D. et al. **Lattice: A Grammar-Based Graphics System**. Journal of Statistical Software, 2008.

SARKAR, D. **Lattice: Multivariate Data Visualization with R**. Springer, 2008. Disponível em: <http://lmdvr.r-forge.r-project.org/>. Acesso em: 30 jun. 2024.

SHIMIZU, G. et al. **AgroR**: Experimental Statistics and Graphics for Agricultural Sciences. [S.l.], 2024. Disponível em: <https://cran.r-project.org/package=AgroR>. Acesso em: 20 ago. 2024.

SILVA, B. **Minicurso de estatística básica**: Introdução ao software R. 2009. Universidade Federal de Santa Maria, Santa Maria.

WARNES, G. **GMODELS**. 2024. Disponível em: <https://cran.r-project.org/web/packages/gmodels/gmodels.pdf>. Acesso em: 30 jun. 2024

WICKHAM, H. et al. **dplyr**: A Grammar of Data Manipulation. [S.l.], 2023. Disponível em: <https://cran.r-project.org/web/packages/dplyr/dplyr.pdf>. Acesso em: 20 ago. 2024.

WICKHAM, H. et al. **Welcome to the tidyverse**. Journal of Open Source Software, 2019. Disponível em: <https://joss.theoj.org/papers/10.21105/joss.01686>. Acesso em: 28 mai. 2024.

WICKHAM, H.; GROLEMUND, G. **R for data science**: import, tidy, transform, visualize, and model data. O'Reilly Media, Inc., 2016. Disponível em: <https://r4ds.hadley.nz/>. Acesso em: 28 mai. 2024.