

CE-214: Planejamento de Experimentos II

Segundo Semestre de 2002

Última atualização: 27 de fevereiro de 2003

Aula 01

Nesta aula é feita uma apresentação geral do curso.

É apresentado o programa computacional R no qual as aulas práticas são baseadas.

Recursos associados ao programa e disponíveis na *web* são apresentados e comentados.

Uma sessão demonstrativa ilustra os aspectos básicos, capacidades e manuseio do programa.

O programa R é gratuito e de código aberto e a página oficial do projeto está em:

<http://www.r-project.org>.

Há também um espelho (*mirror*) brasileiro da área de *downloads* do programa no Departamento de Estatística da UFPR:

<http://www.est.ufpr.br/R>

ou então via FTP em

<ftp://est.ufpr.br/R>

Esta aula utiliza o **RWeb**, um mecanismo que permite rodar o R pela web.

Para acessar o **RWeb** vá até a página do R e no menu à esquerda da página siga os links:

R GUIs . . . R Web

Nesta página selecione primeiro o link **R Web** e examine seu conteúdo. Os participantes do curso são estimulados a explorar os outros recursos da página.

EXERCÍCIOS

1. Na página do **R Web** vá em **R Web modules**, rode e interprete análises com os conjuntos de dados lá disponíveis.
2. Entre com um conjunto de dados seu e use o **R Web** para realizar as análises.

Aula 02

O objetivo desta aula é que os alunos se familiarizem com aspectos básicos do programa R.

Na aula os participantes deverão percorrer todo o conteúdo do tutorial disponível em Tutorial de Introdução ao R.

Ao longo da aula o conteúdo do tutorial deve ser discutido com o professor e demais participantes.

EXERCÍCIO

1. Provavelmente voce não terá tempo de percorrer todo o tutorial durante a aula. Como exercício voce deve examinar cuidadosamente todo o conteúdo.

Aula 03

O objetivo desta aula é mostrar como o programa R pode ser utilizado para análise de experimentos inteiramente casualizados.

Para isto serão utilizados os dados do Exemplo da Tabela 4, página 39 da apostila de *Planejamento de Experimento I* do Prof. Adilson. Clique aqui para visualizar e copiar o arquivo de dados.

Utilize o R para:

- fazer uma análise exploratória dos dados
- montar o quadro de análise de variância utilizando operações aritméticas básicas do R.
- fazer a análise do experimento utilizando as funções `aov()`, `summary()`, `TukeyHSD()` e outras.

Para a análise dos dados voce vai precisar conhecer alguns comandos do R.

Lembre-se que há vários materiais para consulta:

- O “Tutorial de Introdução ao R” contém alguns exemplos de comandos para análise descritiva.
- A página do Rweb contém a sessão **Rweb modules** onde voce pode fazer algumas análises através dos “menus” disponíveis e verificar os resultados para aprender os comandos.
- Lembre-se do **Cartão de Referência** que contém os comandos mais frequentemente utilizados.
- a `demo(graphics)` ilustra comandos para fazer diversos tipos de gráficos.

Aula 04

EXPERIMENTOS INTEIRAMENTE CASUALIZADOS

Nesta aula iremos usar o R para analisar o experimento inteiramente casualizado do exemplo da aula anterior.

A seguir são apresentados os comandos para a análise do experimento. Inspecione-os cuidadosamente e discuta em classe os resultados e a manipulação do programa R.

Primeiro lemos o arquivo de dados que deve ter sido copiado para o seu diretório de trabalho.

```
ex01 <- read.table("exemplo01.txt", head=T)
ex01
```

Caso o arquivo esteja em outro diretório deve-se colocar o caminho completo deste diretório no argumento de `read.table` acima.

A seguir vamos inspecionar o objeto que armazena os dados e suas componentes.:

```
is.data.frame(ex01)
names(ex01)

ex01$resp
ex01$trat

is.factor(ex01$trat)
is.numeric(ex01$resp)
```

Portando concluímos que o objeto é um *data-frame* com duas variáveis, sendo uma delas um fator (a variável *trat*) e a outra uma variável numérica.

Vamos agora fazer uma rápida análise descritiva:

```
summary(ex01)
tapply(ex01$resp, ex01$trat, mean)
```

Há um mecanismo no R de “anexar” objetos ao caminho de procura que permite economizar um pouco de digitação. Veja os comandos abaixo e compara com o comando anterior.

```
search()

attach(ex01)
search()

tapply(resp, trat, mean)
```

Interessante não? Quando “anexamos” um objeto do tipo *list* ou *data.frame* no caminho de procura com o comando `attach()` fazemos com que os componentes deste objeto se tornem imediatamente disponíveis e portanto podemos, por exemplo, digitar somente `trat` ao invés de `ex01$trat`.

Vamos prosseguir com a análise exploratória:

```
ex01.m <- tapply(resp, trat, mean)
ex01.m

ex01.v <- tapply(resp, trat, var)
ex01.v

plot(ex01)
points(ex01.m, pch="x", col=2, cex=1.5)

boxplot(resp ~ trat)
```

Além dos gráficos acima podemos também verificar a homogeneidade de variâncias com o Teste de Bartlett:

```
bartlett.test(resp, trat)
```

Agora vamos fazer a análise de variância. Vamos “desanexar” o objeto com os dados (embora isto não seja obrigatório).

```
detach(ex01)

ex01.av <- aov(resp ~ trat, data = ex01)
ex01.av

summary(ex01.av)
anova(ex01.av)
```

Portanto o objeto `ex01.av` guarda os resultados da análise. Vamos inspecionar este objeto mais cuidadosamente e fazer também uma análise dos resultados e resíduos:

```
names(ex01.av)
ex01.av$coef

ex01.av$res
residuals(ex01.av)

plot(ex01.av) # pressione a tecla enter para mudar o gráfico

par(mfrow=c(2,2))
plot(ex01.av)
par(mfrow=c(1,1))

plot(ex01.av$fit, ex01.av$res, xlab="valores ajustados", ylab="resíduos")
```

```
title("resíduos vs Preditos")

names(anova(ex01.av))
s2 <- anova(ex01.av)$Mean[2]    # estimativa da variância

res <- ex01$res                # extraíndo resíduos
respad <- (res/sqrt(s2))        # resíduos padronizados
boxplot(respad)
title("Resíduos Padronizados" )

hist(respad, main=NULL)
title("Histograma dos resíduos padronizados")

stem(respad)
qqnorm(res,ylab="Residuos", main=NULL)
qqline(res)
title("Grafico Normal de Probabilidade dos Resíduos")

shapiro.test(res)
```

E agora um teste Tukey de comparação múltipla

```
ex01.tu <- TukeyHSD(ex01.av)
plot(ex01.tu)
```

EXERCÍCIO

1. Analise o experimento em blocos ao acaso da apostila de Planejamento de Experimentos I.
Clique aqui para ver e copiar o arquivo de dados.
2. Analise o experimento em quadrado latino da apostila de Planejamento de Experimentos I.
Clique aqui para ver e copiar o arquivo de dados.

Aula 05

Esta aula é dedicada aos experimentos fatoriais.

EXERCÍCIO

1. Analise o experimento fatorial de recipientes da apostila de Planejamento de Experimentos II.

[Clique aqui para ver e copiar o arquivo de dados.](#)

2. Analise o experimento de adubação NPK da apostila de Planejamento de Experimentos II.

[Clique aqui para ver e copiar o arquivo de dados.](#)

Aula 06

EXPERIMENTOS EM ESQUEMA FATORIAL

Nesta aula iremos detalhar e discutir a análise do experimento fatorial do primeiro exercício da aula passada.

Este experimento descrito na apostila do curso de Planejamento de Experimentos II comparou a resposta de mudas a diferentes recipientes e espécies de eucalipto.

No restante destas notas as linhas que começam com o símbolo `>` são comandos a serem digitados no R. Outros textos com a font `typewriter` como `esta` são saídas produzidas pelo programa.

1. Lendo os dados

Clique aqui para ver e copiar o arquivo com conjunto de dados.

A seguir vamos ler (importar) os dados para R com o comando `read.table`:

```
> ex04 <- read.table("exemplo04.txt", header=T)
> ex04
```

Antes de começar as análise vamos inspecionar o objeto que contém os dados para saber quantas observações e variáveis há no arquivo, bem como o nome das variáveis. Vamos também pedir o R que exiba um rápido resumo dos dados.

```
> dim(ex04)
[1] 24  3

> names(ex04)
[1] "rec"  "esp"  "resp"

> attach(ex04)

> is.factor(rec)
[1] TRUE
> is.factor(esp)
[1] TRUE
> is.factor(resp)
[1] FALSE
> is.numeric(resp)
[1] TRUE
```

Nos resultados acima vemos que o objeto `ex04` que contém os dados tem 24 linhas (observações) e 3 colunas (variáveis). As variáveis tem nomes `rec`, `esp` e `resp`, sendo que as duas primeiras são *fatores* enquanto `resp` é uma variável numérica, que neste caso é a

variável resposta. O objeto `ex04` foi incluído no caminho de procura usando o comando `attach` para facilitar a digitação.

2. Análise exploratória

Inicialmente vamos obter um resumo de nosso conjunto de dados usando a função `summary`.

```
> summary(ex04)
rec    esp      resp
r1:8   e1:12  Min.   :18.60
r2:8   e2:12  1st Qu.:19.75
r3:8           Median :23.70
           Mean    :22.97
           3rd Qu.:25.48
           Max.    :26.70
```

Note que para os fatores são exibidos o número de dados em cada nível do fator. Já para a variável numérica são mostrados algumas medidas estatísticas. Vamos explorar um pouco mais os dados

```
> ex04.m <- tapply(resp, list(rec,esp), mean)
> ex04.m
      e1      e2
r1 25.650 25.325
r2 25.875 19.575
r3 20.050 21.325

> ex04.mr <- tapply(resp, rec, mean)
> ex04.mr
      r1      r2      r3
25.4875 22.7250 20.6875

> ex04.me <- tapply(resp, esp, mean)
> ex04.me
      e1      e2
23.85833 22.07500
```

Nos comandos acima calculamos as médias para cada fator, assim como para os cruzamentos entre os fatores. Note que podemos calcular outros resumos além da média. Experimente nos comandos acima substituir `mean` por `var` para calcular a variância de cada grupo, e por `summary` para obter um outro resumo dos dados.

Em experimentos fatoriais é importante verificar se existe interação entre os fatores. Inicialmente vamos fazer isto graficamente e mais a frente faremos um teste formal para

presença de interação. Os comandos a seguir são usados para produzir os gráficos exibidos na Figura 1.

```
> par(mfrow=c(1,2))
> interaction.plot(rec, esp, resp)
> interaction.plot(esp, rec, resp)
```

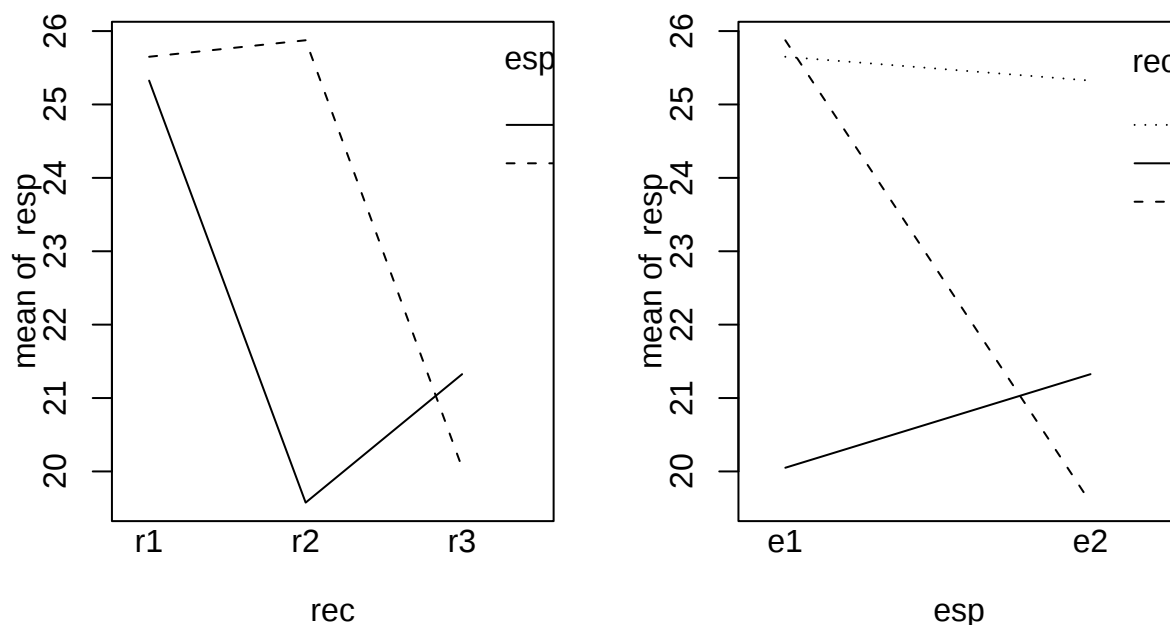


Figura 1: Gráficos de interação entre os fatores.

Pode-se usar o R para obter outros tipos de gráficos de acordo com o interesse de quem está analisando os dados. Por exemplo, os comandos abaixo ilustram outros tipos de gráficos. Experimente estes comandos, verifique os gráficos produzidos e certifique-se que voce entendeu cada comando.

```
> plot.default(rec, resp, ty="n")
> points(rec[esp=="e1"], resp[esp=="e1"], col=1)
> points(ex04.m[,1], pch="x", col=1, cex=1.5)
> points(rec[esp=="e2"], resp[esp=="e2"], col=2)
> points(ex04.m[,2], pch="x", col=2, cex=1.5)

> plot.default(esp, resp, ty="n")
> points(esp[rec=="r1"], resp[rec=="r1"], col=1)
> points(ex04.m[1,], pch="x", col=1, cex=1.5)
> points(esp[rec=="r2"], resp[rec=="r2"], col=2)
> points(ex04.m[2,], pch="x", col=2, cex=1.5)
> points(esp[rec=="r3"], resp[rec=="r3"], col=3)
> points(ex04.m[3,], pch="x", col=3, cex=1.5)
```

```
> coplot(resp ~ rec|esp)
> coplot(resp ~ esp|rec)
```

3. Análise de variância

Seguindo o modelo adequado, o análise de variância para este experimento inteiramente casualizado em esquema fatorial pode ser obtida com o comando:

```
> ex04.av <- aov(resp ~ rec + esp + rec * esp)
```

Entretanto o comando acima pode ser simplificado produzindo os mesmos resultados com o comando

```
> ex04.av <- aov(resp ~ rec * esp)
> summary(ex04.av)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
rec	2	92.861	46.430	36.195	4.924e-07 ***
esp	1	19.082	19.082	14.875	0.001155 **
rec:esp	2	63.761	31.880	24.853	6.635e-06 ***
Residuals	18	23.090	1.283		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Isto significa que no R, ao colocar uma interação no modelo, os efeitos principais são incluídos automaticamente. Note no quadro de análise de variância que a interação é denotada por `rec:esp`. A análise acima mostra que este efeito é significativo, confirmando o que verificamos nos gráficos de interação vistos anteriormente.

O objeto `ex04.av` guarda todos os resultados da análise e pode ser explorado por diversos comandos. Por exemplo a função `model.tables` aplicada a este objeto produz tabelas das médias definidas pelo modelo. O resultado mostra a média geral, médias de cada nível fatores e das combinações dos níveis dos fatores. Note que no resultado está incluído também o número de dados que gerou cada média.

```
> ex04.mt <- model.tables(ex04.av, ty="means")
> ex04.mt
```

Tables of means

Grand mean

22.96667

rec			
	r1	r2	r3
	25.49	22.73	20.69

```
rep 8.00 8.00 8.00
```

```
esp
```

```
    e1    e2
```

```
    23.86 22.07
```

```
rep 12.00 12.00
```

```
rec:esp
```

```
    esp
```

```
rec  e1    e2
```

```
  r1 25.650 25.325
```

```
  rep 4.000 4.000
```

```
  r2 25.875 19.575
```

```
  rep 4.000 4.000
```

```
  r3 20.050 21.325
```

```
  rep 4.000 4.000
```

Mas isto não é tudo! O objeto `ex04.av` possui vários elementos que guardam informações sobre o ajuste.

```
> names(ex04.av)
```

```
[1] "coefficients" "residuals"    "effects"      "rank"
[5] "fitted.values" "assign"        "qr"           "df.residual"
[9] "contrasts"     "xlevels"       "call"         "terms"
[13] "model"
```

```
> class(ex04.av)
```

```
[1] "aov" "lm"
```

O comando `class` mostra que o objeto `ex04.av` pertence às classes `aov` e `lm`. Isto significa que devem haver *métodos* associados a este objeto que tornam a exploração do resultado mais fácil. Na verdade já usamos este fato acima quando digitamos o comando `summary(ex04.av)`. Existe uma função chamada `summary.aov` que foi utilizada já que o objeto é da classe `aov`. Iremos usar mais este mecanismo no próximo passo da análise.

4. Análise de resíduos

Após ajustar o modelo devemos proceder a análise dos resíduos para verificar os pressupostos. O R produz automaticamente 4 gráficos básicos de resíduos conforme a Figura 2 com o comando `plot`.

```
> par(mfrow=c(2,2))
```

```
> plot(ex04.av)
```

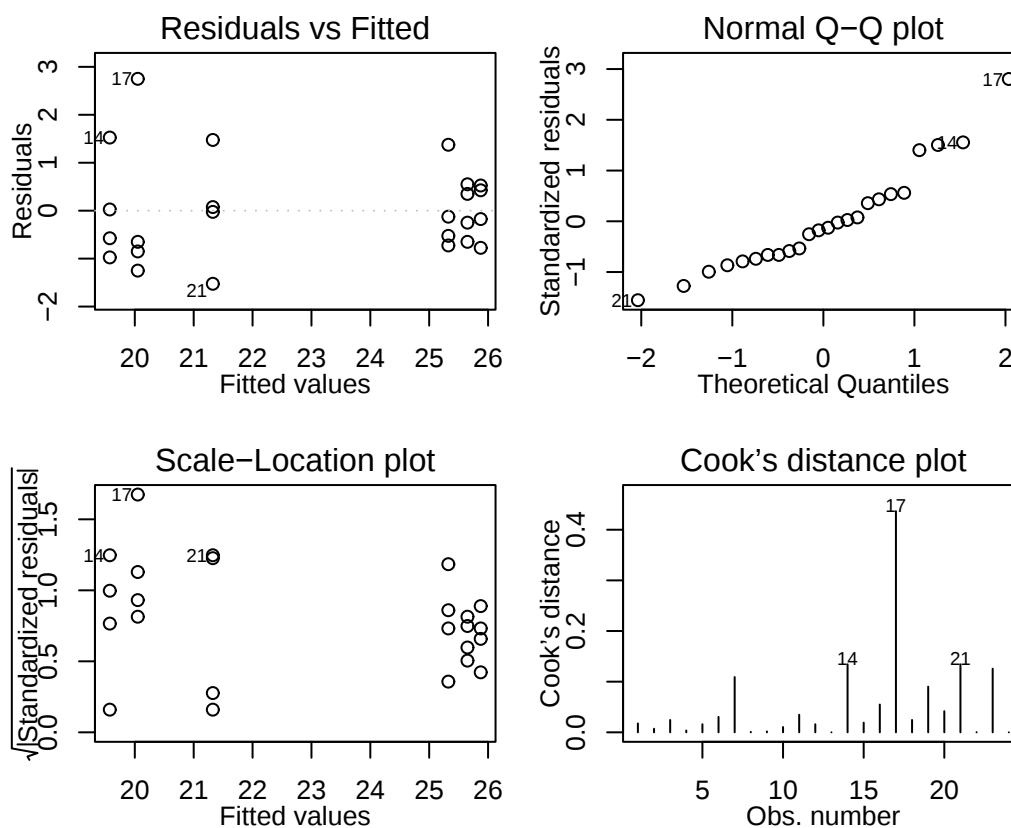


Figura 2: Gráficos de resíduos produzidos automaticamente pelo R.

Os gráficos permitem uma análise dos resíduos que auxiliam no julgamento da adequabilidade do modelo. Evidentemente você não precisa se limitar os gráficos produzidos automaticamente pelo R – você pode criar os seus próprios gráficos muito facilmente. Neste gráficos você pode usar outras variáveis, mudar texto de eixos e títulos, etc, etc, etc. Examine os comandos abaixo e os gráficos por eles produzidos.

```
> par(mfrow=c(2,1))
> residuos <- resid(ex04.av)

> plot(ex04$rec, residuos)
> title("Resíduos vs Recipientes")

> plot(ex04$esp, residuos)
> title("Resíduos vs Espécies")

> par(mfrow=c(2,2))
> preditos <- (ex04.av$fitted.values)
> plot(residuos, preditos)
> title("Resíduos vs Preditos")
> s2 <- sum(resid(ex04.av)^2)/ex04.av$df.res
> respad <- residuos/sqrt(s2)
```

```
> boxplot(respad)
> title("Resíduos Padronizados")
> qqnorm(residuos,ylab="Residuos", main=NULL)
> qqline(residuos)
> title("Grafico Normal de \n Probabilidade dos Resíduos")
```

Além disto há alguns testes já programados. Como exemplo vejamos o teste de Shapiro-Wilk para testar a normalidade dos resíduos.

```
> shapiro.test(residuos)
```

Shapiro-Wilk normality test

```
data:  residuos
W = 0.9293, p-value = 0.09402
```

5. Desdobrando interações

Conforma visto na apostila do curso, quando a interação entre os fatores é significativa podemos desdobrar os graus de liberdade de um fator dentro de cada nível do outro. A forma de fazer isto no R é reajustar o modelo utilizando a notação / que indica efeitos aninhados. Desta forma podemos desdobrar os efeitos de espécie dentro de cada recipiente e vice versa conforme mostrado a seguir.

```
> ex04.avr <- aov(resp ~ rec/esp)
> summary(ex04.avr, split=list("rec:esp"=list(r1=1, r2=2, r3=3)))
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
rec	2	92.861	46.430	36.1952	4.924e-07	***
rec:esp	3	82.842	27.614	21.5269	3.509e-06	***
rec:esp: r1	1	0.211	0.211	0.1647	0.6897	
rec:esp: r2	1	79.380	79.380	61.8813	3.112e-07	***
rec:esp: r3	1	3.251	3.251	2.5345	0.1288	
Residuals	18	23.090	1.283			

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
> ex04.ave <- aov(resp ~ esp/rec)
> summary(ex04.ave, split=list("esp:rec"=list(e1=c(1,3), e2=c(2,4))))
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
esp	1	19.082	19.082	14.875	0.001155	**
esp:rec	4	156.622	39.155	30.524	8.438e-08	***
esp:rec: e1	2	87.122	43.561	33.958	7.776e-07	***

```

    esp:rec: e2  2  69.500  34.750  27.090 3.730e-06 ***
Residuals    18  23.090   1.283
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

6. Teste de Tukey para comparações múltiplas

Há vários testes de comparações múltiplas disponíveis na literatura, e muitos deles implementados no R. Os que não estão implementados podem ser facilmente calculados utilizando os recursos do R.

Vejamos por exemplo duas formas de usar o *Teste de Tukey*, a primeira usando uma implementação com a função `TukeyHSD` e uma segunda fazendo ops cálculos necessários com o R.

Poderíamos simplesmente digitar:

```

> ex04.tk <- TukeyHSD(ex04.av)
> plot(ex04.tk)
> ex04.tk

```

e obter diversos resultados. Entretanto nem todos nos interessam. Como a interação foi significativa na análise deste experimento a comparação dos níveis fatores principais não nos interessa.

Podemos então pedir a função que somente mostre a comparação de médias entre as combinações dos níveis dos fatores.

```

> ex04.tk <- TukeyHSD(ex04.ave, "esp:rec")
> plot(ex04.tk)
> ex04.tk
    Tukey multiple comparisons of means
    95% family-wise confidence level

```

```

Fit: aov(formula = resp ~ esp/rec)

```

```

$"esp:rec"
      diff      lwr      upr
[1,] -0.325 -2.8701851  2.220185
[2,]  0.225 -2.3201851  2.770185
[3,] -6.075 -8.6201851 -3.529815
[4,] -5.600 -8.1451851 -3.054815
[5,] -4.325 -6.8701851 -1.779815
[6,]  0.550 -1.9951851  3.095185
[7,] -5.750 -8.2951851 -3.204815

```

```

[8,] -5.275 -7.8201851 -2.729815
[9,] -4.000 -6.5451851 -1.454815
[10,] -6.300 -8.8451851 -3.754815
[11,] -5.825 -8.3701851 -3.279815
[12,] -4.550 -7.0951851 -2.004815
[13,]  0.475 -2.0701851  3.020185
[14,]  1.750 -0.7951851  4.295185
[15,]  1.275 -1.2701851  3.820185

```

Mas ainda assim temos resultados que não interessam. Mais especificamente estamos interessados nas comparações dos níveis de um fator dentro dos níveis de outro. Por exemplo, vamos fazer as comparações dos recipientes para cada uma das espécies.

Primeiro vamos obter

```

> s2 <- sum(resid(ex04.av)^2)/ex04.av$df.res
> dt <- qtkey(0.95, 3, 18) * sqrt(s2/4)
> dt
[1] 2.043945
>
> ex04.m
      e1      e2
r1 25.650 25.325
r2 25.875 19.575
r3 20.050 21.325
>
> m1 <- ex04.m[,1]
> m1
      r1      r2      r3
25.650 25.875 20.050
> m1d <- outer(m1,m1,"-")
> m1d
      r1      r2      r3
r1  0.000 -0.225  5.600
r2  0.225  0.000  5.825
r3 -5.600 -5.825  0.000
> m1d <- m1d[lower.tri(m1d)]
> m1d
      r2      r3    <NA>
0.225 -5.600 -5.825
>
> m1n <- outer(names(m1),names(m1),paste, sep="-")
> names(m1d) <- m1n[lower.tri(m1n)]

```

```
> m1d
  r2-r1  r3-r1  r3-r2
0.225 -5.600 -5.825
>
> data.frame(dif = m1d, sig = ifelse(abs(m1d) > dt, "*", "ns"))
      dif sig
r2-r1 0.225 ns
r3-r1 -5.600 *
r3-r2 -5.825 *
>
> m2 <- ex04.m[,2]
> m2d <- outer(m2,m2,"-")
> m2d <- m2d[lower.tri(m2d)]
> m2n <- outer(names(m2),names(m2),paste, sep="-")
> names(m2d) <- m2n[lower.tri(m2n)]
> data.frame(dif = m2d, sig = ifelse(abs(m2d) > dt, "*", "ns"))
      dif sig
r2-r1 -5.75  *
r3-r1 -4.00  *
r3-r2  1.75 ns
```

Aula 07

EXPERIMENTOS HIERÁRQUICOS

Nesta aula veremos como analisar experimentos hierárquicos no R.

Vamos considerar primeiro o exemplo da apostila retirado do livro de Montgomery. Clique aqui para ver e copiar o arquivo com conjunto de dados.

O experimento estuda a variabilidade de lotes e fornecedores na puraza da matéria prima. A análise assume que os fornecedores são um efeito fixo enquanto que lotes são efeitos aleatórios.

Inicialmente vamos ler (importar) os dados para R com o comando `read.table`. A seguir vamos examinar o objeto que contém os dados.

```
> ex06 <- read.table("exemplo06.txt", header=T)
> ex06
```

```
> dim(ex06)
[1] 36  3
```

```
> names(ex06)
[1] "forn" "lot"  "resp"
```

```
> is.factor(ex06$forn)
[1] FALSE
> is.factor(ex06$lot)
[1] FALSE
```

```
> ex06$forn <- as.factor(ex06$forn)
> ex06$lot <- as.factor(ex06$lot)
```

```
> is.factor(ex06$resp)
[1] FALSE
> is.numeric(ex06$resp)
[1] TRUE
```

```
> summary(ex06)
forn   lot      resp
1:12   1:9   Min.    :-4.0000
2:12   2:9   1st Qu.: -1.0000
3:12   3:9   Median  : 0.0000
        4:9   Mean    : 0.3611
        3rd Qu.: 2.0000
        Max.    : 4.0000
```

Nos comandos acima verificamos que o objeto `ex06` possui 36 linhas correspondentes às

observações e 3 colunas que correspondem às variáveis **forn** (fornecedor), **lot** (lote) e **resp** (a variável resposta).

A seguir verificamos que **forn** e **lot** não foram lidas como fatores. **NÃO** podemos seguir as análise desta forma pois o R leria os valores destas variáveis como quantidades numéricas e não como indicadores dos níveis dos fatores. Para corrigir isto usamos o comando **as.factor** para indicar ao R que estas variáveis são fatores.

Finalmente verificamos que a variável resposta é numérica e produzimos um rápido resumo dos dados.

Na sequência deveríamos fazer uma análise exploratória, alguns gráficos descritivos etc, como na análise do experimento da aula anterior. Vamos deixar isto cpor conta do leitor e passar direto para a análise de variância.

A notação para indicar efeitos aninhados no modelo é /. Desta forma poderíamos ajustar o modelo da seguinte forma:

```
> ex06.av <- aov(resp ~ forn/lot, data=ex06)
> summary(ex06.av)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
forn	2	15.056	7.528	2.8526	0.07736 .
forn:lot	9	69.917	7.769	2.9439	0.01667 *
Residuals	24	63.333	2.639		

```
---
```

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Embora os elementos do quadro de análise de variância estejam corretos o teste F para efeito dos fornecedores está **ERRADO**. A análise acima considerou todos os efeitos como fixos e portanto dividiu os quadrados médios dos efeitos pelo quadrado médio do resíduo. Como **lotes** é um efeito aleatório deveríamos dividir o quadrado médio de to termo **lot** pelo quadrado médio de **forn:lot**

Uma forma de indicar a estrutura hierárquica ao R é especificar o modelo de forma que o termo de resíduo seja dividido de maneira adequada. Veja o resultado abaixo.

```
> ex06.av1 <- aov(resp ~ forn/lot + Error(forn) , data=ex06)
> summary(ex06.av1)
```

Error: forn

	Df	Sum Sq	Mean Sq
forn	2	15.0556	7.5278

Error: Within

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
forn:lot	9	69.917	7.769	2.9439	0.01667 *
Residuals	24	63.333	2.639		

```
---
```

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Agora o teste F errado não é mais mostrado, mas o teste correto também não foi feito! Isto não é problema porque podemos extrair os elementos que nos interessam e fazer o teste desejado. Primeiro vamos guardar verificamos que o comando **anova** produz uma lista que tem entre seus elementos os graus de liberdade **Df** e os quadrados médios (**Mean Sq**). A partir destes elementos podemos obtemos o valor da estatística F e o valor P associado.

```
> ex06.anova <- anova(ex06.av)
> is.list(ex06.anova)
[1] TRUE

> names(ex06.anova)
[1] "Df"          "Sum Sq"      "Mean Sq"     "F value"     "Pr(>F)"

> ex06.anova$Df
 1  2
 2  9 24
> ex06.anova$Mean
      1      2
7.527778 7.768519 2.638889

> Fcalc <- ex06.anova$Mean[1]/ex06.anova$Mean[2]
> Fcalc
      1
0.9690107
> pvalor <- 1 - pf(Fcalc, ex06.anova$Df[1], ex06.anova$Df[2])
> pvalor
      1
0.4157831
```

USANDO O PACOTE NLME

Uma outra possível e elegante solução no R para este problema é utilizar a função **lme** do pacote **nlme**. Note que a abordagem do problema por este pacote é um pouco diferente da forma apresentada no curso por se tratar de uma ferramenta geral para modelos com efeitos aleatórios. Entretanto os todos os elementos relevantes da análise estão incluídos nos resultados. Vamos a seguir ver os comandos necessários comentar os resultados.

Inicialmente temos que carregar o pacote **nlme** com o comando **require**.

A seguir criamos uma variável para indicar o efeito aleatório que neste exemplo chamamos de **ex06\$fa** utilizando a função **getGroups**.

Feito isto podemos rodar a função **lme** que faz o ajuste do modelo.

```
> require(nlme)
[1] TRUE
```

```
> ex06$fa <- getGroups(ex06, ~ 1|forn/lot, level=2)
> ex06.av <- lme(resp ~ forn, random=~1|fa, data=ex06)
> ex06.av
```

Linear mixed-effects model fit by REML

Data: ex06

Log-restricted-likelihood: -71.42198

Fixed: resp ~ forn

(Intercept)	forn2	forn3
-0.4166667	0.7500000	1.5833333

Random effects:

Formula: ~1 | fa

(Intercept) Residual

StdDev: 1.307561 1.624483

Number of Observations: 36

Number of Groups: 12

Este modelo tem a variável **forn** como efeito fixo e a variável **lot** como efeito aleatório com o componente de variância σ_{lote}^2 . Além disto temos a variância residual σ^2 . A saída acima mostra as estimativas destes componentes da variância como sendo $\hat{\sigma}_{lote}^2 = (1.307)^2 = 1.71$ e $\hat{\sigma}^2 = (1.624)^2 = 2.64$.

O comando **anova** vai mostrar a análise de variância com apenas os efeitos principais. O fato do programa não incluir o efeito aleatório de lotes na saída não causa problema algum. O comando **intervals** mostra os intervalos de confiança para os componentes de variância. Portanto para verificar a significância do efeito de lotes basta ver se o intervalo para este componente de variância exclui o valor 0, o que é o caso neste exemplo conforme vamos abaixo.

```
> anova(ex06.av)
              numDF denDF    F-value p-value
(Intercept)      1     24 0.6043242  0.4445
forn              2      9 0.9690643  0.4158
> intervals(ex06.av)
Approximate 95% confidence intervals
```

Fixed effects:

	lower	est.	upper
(Intercept)	-2.0772277	-0.4166667	1.243894
forn2	-1.8239746	0.7500000	3.323975
forn3	-0.9906412	1.5833333	4.157308

Random Effects:

Level: fa

```

              lower      est.      upper
sd((Intercept)) 0.6397003 1.307561 2.672682

```

Within-group standard error:

```

      lower      est.      upper
1.224202 1.624483 2.155644

```

Finalmente uma versão mais detalhada dos resultados pode ser obtida com o comando `summary`.

```
> summary(ex06.av)
```

Linear mixed-effects model fit by REML

Data: ex06

```

      AIC      BIC    logLik
152.8440 160.3265 -71.42198

```

Random effects:

Formula: ~1 | fa

(Intercept) Residual

```
StdDev:    1.307561 1.624483
```

Fixed effects: resp ~ forn

	Value	Std.Error	DF	t-value	p-value
(Intercept)	-0.4166667	0.8045749	24	-0.5178718	0.6093
forn2	0.7500000	1.1378407	9	0.6591432	0.5263
forn3	1.5833333	1.1378407	9	1.3915246	0.1975

Correlation:

(Intr) forn2

forn2 -0.707

forn3 -0.707 0.500

Standardized Within-Group Residuals:

	Min	Q1	Med	Q3	Max
	-1.4751376	-0.7500844	0.0812409	0.7060895	1.8720268

Number of Observations: 36

Number of Groups: 12

O próximo passo da seria fazer uma análise dos resíduos para verificar os pressupostos, semelhante ao que foi feito no experimento fatorial da aula anterior. Vamos deixar isto por conta do leitor.

EXERCÍCIO

1. Faça uma análise completa do experimento hierárquico descrito acima.
2. Faça a análise do seguinte experimento descrito na apostila do curso, conduzido no arranjo hierárquico, com 3 repetições.

Tabela 1: Diâmetro de peças produzidas por diferentes máquinas e operadores (mm).

	Máquina 1		Máquina 2		Máquina 3		Máquina 4	
Operador	1	2	1	2	1	2	1	2
Amostra 1	30,66	30,85	30,97	30,67	31,03	30,76	30,77	31,03
Amostra 2	30,78	31,00	30,97	30,75	31,01	30,69	30,69	31,09
Amostra 3	30,82	30,87	30,95	30,81	30,93	30,69	30,71	30,89

EXPERIMENTOS COM FATORES HIERÁRQUICOS E CRUZADOS

Agora vejamos como montar a análise do experimento sobre peso de aves, com fatores cruzados e hierárquicos descrito na apostila do curso. Iremos mostrar apenas a obtenção da análise de variância, ficando a análise exploratória, de resíduos e comparações múltiplas por conta do leitor.

Clique aqui para ver e copiar o arquivo com conjunto de dados. Vamos a seguir ler e preparar os dados no R.

```
> ex08 <- read.table("exemplo08.txt", header=T)
> ex08
```

```
> dim(ex08)
[1] 48 4
> names(ex08)
[1] "emp" "criad" "prod" "resp"
>
> ex08$emp <- as.factor(ex08$emp)
> ex08$criad <- as.factor(ex08$criad)
> ex08$prod <- as.factor(ex08$prod)
>
> summary(ex08)
emp      criad  prod      resp
1:24    1:12   1:16   Min.    :21.00
2:24    2:12   2:16   1st Qu.:24.00
        3:12   3:16   Median :26.00
        4:12           Mean    :26.08
                        3rd Qu.:28.00
                        Max.    :32.00
```

Assim como no exemplo anterior a análise com o modelo especificado abaixo produz o quadro de análise de variância com os elementos, e quadrados médios corretos porém com valores F **ERRADOS**. Se usarmos o termo **Error** a saída não inclui os testes incorretos.

```
> ex08.av <- aov(resp ~ prod*(emp/criad), data=ex08)
> summary(ex08.av)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
prod	2	82.792	41.396	17.7411	1.862e-05 ***
emp	1	4.083	4.083	1.7500	0.198344
prod:emp	2	19.042	9.521	4.0804	0.029828 *
emp:criad	6	71.917	11.986	5.1369	0.001606 **
prod:emp:criad	12	65.833	5.486	2.3512	0.036043 *
Residuals	24	56.000	2.333		

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> ex08.av1 <- aov(resp ~ prod*(emp/criad) + Error(prod*emp), data=ex08)
> summary(ex08.av1)
```

```
Error: prod
      Df Sum Sq Mean Sq
prod   2 82.792  41.396
```

```
Error: emp
      Df Sum Sq Mean Sq
emp    1  4.083   4.083
```

```
Error: prod:emp
      Df Sum Sq Mean Sq
prod:emp 2 19.0417  9.5208
```

```
Error: Within
      Df Sum Sq Mean Sq F value    Pr(>F)
emp:criad      6 71.917  11.986  5.1369 0.001606 **
prod:emp:criad 12 65.833   5.486  2.3512 0.036043 *
Residuals     24 56.000   2.333
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Portanto, para testar os fatores que faltam extraímos as quantidades necessárias e fazemos os testes F com os denominadores corretos.

```
> ex08.anova <- anova(ex08.av)
```

```

> is.list(ex08.anova)
[1] TRUE
> names(ex08.anova)
[1] "Df"      "Sum Sq"  "Mean Sq" "F value" "Pr(>F)"

> ex08.anova$Df
  1  2  3  4  5
  2  1  2  6 12 24
> ex08.anova$Mean
      1      2      3      4      5
41.395833  4.083333  9.520833 11.986111  5.486111  2.333333
>
> Fprod <- ex08.anova$Mean[1]/ex08.anova$Mean[5]
> pv.prod <- 1 - pf(Fprod, ex08.anova$Df[1], ex08.anova$Df[5])
> c(Fprod = Fprod, Pvalor = pv.prod)
      Fprod.1    Pvalor.1
7.545569620 0.007553076

> Femp <- ex08.anova$Mean[2]/ex08.anova$Mean[4]
> pv.emp <- 1 - pf(Femp, ex08.anova$Df[2], ex08.anova$Df[4])
> c(Femp = Femp, Pvalor = pv.emp)
      Femp.2    Pvalor.2
0.3406721 0.5807041

> Fpe <- ex08.anova$Mean[3]/ex08.anova$Mean[5]
> pv.pe <- 1 - pf(Fpe, ex08.anova$Df[3], ex08.anova$Df[5])
> c(Fpe = Fpe, Pvalor = pv.pe)
      Fpe.3    Pvalor.3
1.7354430 0.2177691

```

EXERCÍCIO

1. Complete a análise do experimento acima com a verificação dos pressupostos e desdobramentos de interações e comparações múltiplas (se for recomendado para este caso).

Aula 08

EXPERIMENTOS EM PARCELAS SUBDIVIDIDAS E EM FAIXAS

Vamos mostrar aqui como especificar o modelo para análise de experimentos em parcelas subdivididas e em faixas. Os comandos abaixo mostram a leitura e preparação dos dados e a obtenção da análise de variância. Deixamos por conta do leitor a análise exploratória e de resíduos, desdobramento das interações e testes de comparações múltiplas.

1. Parcelas subdivididas

Considere o experimento em parcelas subdivididas de dados de produção de aveia descrito na apostila do curso. Clique aqui para ver e copiar o arquivo com conjunto de dados. A obtenção da análise de variância é ilustrada nos comandos e saídas abaixo.

```
> ex09 <- read.table("exemplo09.txt", header=T)
> ex09

> dim(ex09)
[1] 64  4
> names(ex09)
[1] "a"      "b"      "bloco" "resp"
>
> ex09$a <- as.factor(ex09$a)
> ex09$b <- as.factor(ex09$b)
> ex09$bloco <- as.factor(ex09$bloco)
>
> summary(ex09)
  a      b      bloco      resp
1:16  1:16  1:16  Min.   :28.30
2:16  2:16  2:16  1st Qu.:44.90
3:16  3:16  3:16  Median :52.30
4:16  4:16  4:16  Mean    :52.81
                     3rd Qu.:62.38
                     Max.    :75.40
>
> ex09.av <- aov(resp ~ bloco + a*b + Error(bloco/a), data=ex09)
> summary(ex09.av)

Error: bloco
      Df Sum Sq Mean Sq
bloco  3 2842.87  947.62

Error: bloco:a
```

```

          Df Sum Sq Mean Sq F value    Pr(>F)
a           3 2848.02   949.34   13.819 0.001022 **
Residuals   9   618.29    68.70
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Error: Within
          Df Sum Sq Mean Sq F value    Pr(>F)
b           3  170.54    56.85   2.7987 0.053859 .
a:b          9  586.47    65.16   3.2082 0.005945 **
Residuals  36  731.20    20.31
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

2. Faixas

Considere o experimento em faixas de dados de produção de milho descrito na apostila do curso. Clique aqui para ver e copiar o arquivo com conjunto de dados. A obtenção da análise de variância é ilustrada nos comandos e saídas abaixo.

```

> ex10 <- read.table("exemplo10.txt", header=T)
> ex10

> dim(ex10)
[1] 48  4
> names(ex10)
[1] "espA" "denB" "bloco" "resp"

> ex10$espA <- as.factor(ex10$espA)
> ex10$denB <- as.factor(ex10$denB)
> ex10$bloco <- as.factor(ex10$bloco)

> summary(ex10)
espA   denB   bloco      resp
1:12   1:16   1:12   Min.    :16.38
2:12   2:16   2:12   1st Qu.:17.20
3:12   3:16   3:12   Median :17.50
4:12           4:12   Mean    :17.44
                        3rd Qu.:17.68
                        Max.    :18.27

> ex10.av <- aov(resp ~ bloco + espA*denB + Error(bloco/(espA+denB)), data=ex10)
> summary(ex10.av)

```

Error: bloco

	Df	Sum Sq	Mean Sq
bloco	3	0.99206	0.33069

Error: bloco:espA

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
espA	3	0.67532	0.22511	1.2527	0.3473
Residuals	9	1.61729	0.17970		

Error: bloco:denB

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
denB	2	0.27691	0.13846	0.3622	0.7104
Residuals	6	2.29354	0.38226		

Error: Within

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
espA:denB	6	0.61647	0.10275	0.9469	0.4869
Residuals	18	1.95315	0.10851		

EXERCÍCIO

1. Faça uma análise completa do experimento de aveia em parcelas subdivididas descrito acima.
2. Faça uma análise completa do experimento de milho em faixas descrito acima.

Aula 09

ANÁLISE DE GRUPOS DE EXPERIMENTOS

Vejamos agora a análise de grupos de experimentos. Para isto vamos considerar o exemplo de competição de cultivares de milho mostrado na apostila do curso.

Clique aqui para ver e copiar o arquivo com conjunto de dados. Como já estamos acostumados começando lendo e preparando os dados.

```
> ex11 <- read.table("dados/exemplo11.txt", header=T)
> ex11

> dim(ex11)
[1] 80 4
> names(ex11)
[1] "local"    "bloco"    "trat"     "resposta"

> ex11$local <- as.factor(ex11$local)
> ex11$bloco <- as.factor(ex11$bloco)
> ex11$trat <- as.factor(ex11$trat)

> summary(ex11)
  local  bloco  trat      resposta
1:20   1:20   1:16   Min.      :1.700
2:20   2:20   2:16   1st Qu.:2.275
3:20   3:20   3:16   Median  :2.900
4:20   4:20   4:16   Mean     :2.723
              5:16   3rd Qu.:3.125
              Max.     :3.500
```

Nesta análise a primeira coisa a se fazer é a análise de cada experimento individualmente. Para isto vamos particionar os nossos dados usando o comando `split` e obter um resumo de cada experimento.

```
> ex11.1 <- split(ex11, ex11$local)

> lapply(ex11.1, summary)
$"1"
  local  bloco  trat      resposta
1:20   1:5    1:4   Min.      :1.90
2: 0    2:5    2:4   1st Qu.:2.35
3: 0    3:5    3:4   Median  :2.85
4: 0    4:5    4:4   Mean     :2.74
              5:4   3rd Qu.:3.10
```

Max. :3.40

\$"2"

local	bloco	trat	resposta
1: 0	1:5	1:4	Min. :1.800
2:20	2:5	2:4	1st Qu.:2.275
3: 0	3:5	3:4	Median :2.850
4: 0	4:5	4:4	Mean :2.730
		5:4	3rd Qu.:3.200
			Max. :3.500

\$"3"

local	bloco	trat	resposta
1: 0	1:5	1:4	Min. :1.700
2: 0	2:5	2:4	1st Qu.:2.300
3:20	3:5	3:4	Median :2.850
4: 0	4:5	4:4	Mean :2.735
		5:4	3rd Qu.:3.225
			Max. :3.400

\$"4"

local	bloco	trat	resposta
1: 0	1:5	1:4	Min. :1.800
2: 0	2:5	2:4	1st Qu.:2.175
3: 0	3:5	3:4	Median :2.900
4:20	4:5	4:4	Mean :2.685
		5:4	3rd Qu.:3.100
			Max. :3.400

No resumo acima percebemos que todos os experimentos possuem a mesma estrutura, isto é, são todos em delineamento de blocos casualizados com 4 blocos e 5 tratamentos. A seguir vamos obter a análise de cada experimento individualmente. Como os experimentos possuem a mesma estrutura isto pode ser feito de usando um *loop* como se segue.

```
> ex11.lav <- list()
> for(i in names(ex11.l))
+   ex11.lav[[i]] <- anova(aov(resposta ~ bloco + trat, data=ex11.l[[i]]))
```

```
> ex11.lav
```

\$"1"

Analysis of Variance Table

Response: resposta

```

      Df Sum Sq Mean Sq F value    Pr(>F)
bloco    3  0.0080   0.0027   0.0732    0.9732
trat     4  3.8830   0.9707  26.6568 6.85e-06 ***
Residuals 12  0.4370   0.0364

```

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
$"2"
```

```
Analysis of Variance Table
```

```
Response: resposta
```

```

      Df Sum Sq Mean Sq F value    Pr(>F)
bloco    3  0.1020   0.0340   0.9645    0.4411
trat     4  5.1570   1.2892  36.5745 1.242e-06 ***
Residuals 12  0.4230   0.0353

```

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
$"3"
```

```
Analysis of Variance Table
```

```
Response: resposta
```

```

      Df Sum Sq Mean Sq F value    Pr(>F)
bloco    3  0.0455   0.0152   0.8387    0.4984
trat     4  5.4230   1.3557  74.9724 2.196e-08 ***
Residuals 12  0.2170   0.0181

```

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
$"4"
```

```
Analysis of Variance Table
```

```
Response: resposta
```

```

      Df Sum Sq Mean Sq F value    Pr(>F)
bloco    3  0.0255   0.0085   0.375    0.7727
trat     4  4.8880   1.2220  53.912 1.434e-07 ***
Residuals 12  0.2720   0.0227

```

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Portanto o objeto `ex11.lav` guarda os resultados da análise de cada experimento. Para verificar a homogeneidade entre os experimento devemos extrair os quadrados médios de cada

experimento e fazer o teste de Hartley dividindo o maior pelo menor valor de quadrado médio.

```
> ex11.lqm <- numeric()
> for(i in names(ex11.l))
+   ex11.lqm[i] <- anova(aov(resposta ~ bloco + trat, data=ex11.l[[i]]))$Mean[3]
> ex11.lqm
      1      2      3      4
0.03641667 0.03525000 0.01808333 0.02266667

> max(ex11.lqm)/min(ex11.lqm)
[1] 2.013825
```

O valor acima comparado com o valor tabelado da tabela de Hartley mostra que não há diferença significativa entre as variâncias. Portanto podemos prosseguir fazendo a análise conjunta do experimento.

```
> ex11.av <- aov(resposta ~ local/bloco + trat/local, data=ex11)
> summary(ex11.av)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
local	3	0.0385	0.0128	0.4566	0.7138
trat	4	19.1495	4.7874	170.3440	<2e-16 ***
local:bloco	12	0.1810	0.0151	0.5367	0.8797
local:trat	12	0.2015	0.0168	0.5975	0.8333
Residuals	48	1.3490	0.0281		

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Como em exemplos anteriores o quadro da análise de variância acima possui os elementos corretos porém alguns dos testes F estão **ERRADOS** por não considerar que certos efeitos são aleatórios. Como já vimos isto pode ser corrigido facilmente extraindo os elementos necessários para testar o efeito desejado. Neste caso, para testar o efeito dos tratamentos basta executar os comandos mostrados abaixo.

```
> ex11.anova <- anova(ex11.av)

> Ftrat <- ex11.anova$Mean[2]/ex11.anova$Mean[4]
> pv.trat <- 1 - pf(Ftrat, ex11.anova$Df[2], ex11.anova$Df[4])
> c(Ftrat = Ftrat, pvalor = pv.trat)
```

Ftrat.2	pvalor.2
2.851042e+02	8.843704e-12

EXERCÍCIO

1. Faça uma análise mais cuidadosa do experimento vista acima, verificando os resíduos e fazendo comparações múltiplas.

2. Faça uma análise completa do experimento de grupos proposto na apostila do curso.

Aula 10

ANÁLISE DE COVARIÂNCIA

Vejam agora a análise de covariância do exemplo da apostila do curso. Clique aqui para ver e copiar o arquivo com conjunto de dados. Começamos com a leitura e organização dos dados. Note que neste caso temos 2 variáveis numéricas, a resposta (*resp*) e a covariável (*cov*).

```
> ex12 <- read.table("dados/exemplo12.txt", header=T)
> ex12

> dim(ex12)
[1] 15 3
> names(ex12)
[1] "maq" "cov" "resp"
>
> ex12$maq <- as.factor(ex12$maq)
> is.numeric(ex12$cov)
[1] TRUE
> is.numeric(ex12$resp)
[1] TRUE
>
> summary(ex12)
```

	maq	cov	resp
1:5	Min.	:15.00	Min. :32.0
2:5	1st Qu.:	21.50	1st Qu.:36.5
3:5	Median	:24.00	Median :40.0
	Mean	:24.13	Mean :40.2
	3rd Qu.:	27.00	3rd Qu.:43.0
	Max.	:32.00	Max. :49.0

Na análise de covariância os testes de significância tem que ser obtidos em ajustes separados. Isto é porque não temos ortogonalidade entre os fatores.

Primeiro vamos testar o intercepto (coeficiente β) da reta de regressão. Na análise de variância abaixo devemos considerar apenas o teste referente à variável *cov* que neste caso está corrigida para o efeito de *maq*. Note que para isto a variável *cov* tem que ser a última na especificação do modelo.

```
> ex12.av <- aov(resp ~ maq + cov, data=ex12)
> summary(ex12.av)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
maq	2	140.400	70.200	27.593	5.170e-05 ***
cov	1	178.014	178.014	69.969	4.264e-06 ***
Residuals	11	27.986	2.544		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

A seguir testamos o efeito do fator `maq` corrigindo para o efeito da covariável. Para isto basta inverter a ordem dos termos na especificação do modelo.

```
> ex12.av <- aov(resp ~ cov + maq, data=ex12)
```

```
> summary(ex12.av)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
cov	1	305.130	305.130	119.9330	2.96e-07	***
maq	2	13.284	6.642	2.6106	0.1181	
Residuals	11	27.986	2.544			

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Solução dos Exercícios

O arquivo `ce214pratica.R` contém os comandos do R para resolver exercícios propostos nas aulas práticas.