

Aulas Práticas

CE074: Controle de Processos Industriais

Adilson dos Anjos

última atualização: 2 de setembro de 2008

Aula 01: Experimentos em esquema fatorial

Objetivo dessa aula

O objetivo dessa aula é realizar a análise de variância de experimentos fatoriais. Será utilizado o Software R.

Nesta aula iremos detalhar e discutir a análise do experimento fatorial 2×2 e 2×3 .

Procure entender os comandos utilizados e também interpretar os resultados.

Trabalhando com o arquivo de dados

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.

Experimento Fatorial 2^2

Lendo os dados

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

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

```
> ex01 <- read.table("fat22.txt", header=T)
> ex01
```

Lembre-se de fornecer o caminho (onde você salvou os dados).

Antes de começar as análises 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 para o R que exiba um rápido resumo dos dados.

```
> dim(ex01)

> names(ex01)

> attach(ex01)

> is.factor(A)

> is.factor(B)
```

```
> is.factor(ESP)
```

```
> is.numeric(ESP)
```

Caso alguma resposta não esteja de acordo com o necessário, isso deve ser corrigido.

O objeto `ex01` foi incluído no caminho de procura usando o comando `attach` para facilitar a digitação.

Análise descritiva

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

```
> summary(ex01)
```

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

```
> ex01.m <- tapply(ESP, list(A,B), mean)
```

```
> ex01.m
```

```
> ex01.mA <- tapply(ESP, A, mean)
```

```
> ex01.mA
```

```
> ex01.mB <- tapply(ESP, B, mean)
```

```
> ex01.mB
```

Nos comandos acima calculamos as médias para cada fator, assim como para os cruzamentos entre os fatores.

Note que podemos calcular outras estatísticas além da média. Experimente outras medidas!

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.

```
> par(mfrow=c(1,2))
```

```
> interaction.plot(A, B, ESP)
```

```
> interaction.plot(B, A, ESP)
```

Análise de variância

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

```
> ex01.av <- aov(RESPOSTA ~ A + B + A * B)
```

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

```
> ex01.av <- aov(RESPOSTA ~ A * B)
> summary(ex01.av)
```

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 A:B.

A análise acima mostra que este efeito é não significativo, confirmando o que verificamos nos gráficos de interação vistos anteriormente.

O objeto `ex01.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.

```
> ex01.mt <- model.tables(ex01.av, ty="means")
> ex01.mt
```

Mas isto não é tudo! O objeto `ex01.av` possui vários elementos que guardam informações sobre o ajuste.

```
> names(ex01.av)
[1] "coefficients" "residuals"      "effects"         "rank"
[5] "fitted.values" "assign"          "qr"              "df.residual"
[9] "contrasts"     "xlevels"         "call"            "terms"
[13] "model"

> class(ex01.av)
[1] "aov" "lm"
```

O comando `class` mostra que o objeto `ex01.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(ex01.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.

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 com o comando `plot`.

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

Os gráficos permitem uma análise dos resíduos que auxiliam no julgamento da adequabilidade do modelo. Evidentemente você não precisa se limitar aos 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. Examine os comandos abaixo e os gráficos por eles produzidos.

```
> par(mfrow=c(2,1))
> residuos <- resid(ex01.av)

> plot(ex01$A, residuos)
> title("Resíduos vs Fator A")

> plot(ex01$B, residuos)
> title("Resíduos vs Fator B")

> par(mfrow=c(2,2))
> preditos <- (ex01.av$fitted.values)
> plot(residuos, preditos)
> title("Resíduos vs Preditos")
> s2 <- sum(resid(ex01.av)^2)/ex01.av$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)
```

Teste de Tukey para comparações múltiplas

```
> TukeyHSD(ex01.av)
```

Observe que apenas o teste para o fator A e B são de interesse.

Experimento fatorial 2^3

Objetivo do exercício

O objetivo desse exercício é realizar a análise de variância de um experimento fatorial com três fatores, utilizando o software R.

Trabalhando com o arquivo de dados

Nesse exercício será analisado o experimento fatorial 2^3 conduzido em um delineamento em blocos completos casualizados apresentado nas notas de aula.

O arquivo de dados pode ser obtido aqui..

```
ex02<-read.table("ex02.txt",header=T)
```

Lembre-se que cada coluna indicadora dos fatores e blocos deve ser um fator no R. Por essa razão, os vetores A, B, C e bloco precisam ser transformados para fatores.

```
> is.factor(ex02$A)
[1] FALSE
> is.factor(ex02$B)
[1] FALSE
> is.factor(ex02$C)
[1] FALSE
> is.factor(ex02$bloco)
[1] FALSE
```

Para facilitar a digitação, pode-se incluir no caminho de procura o objeto `ex02` com o comando `attach(ex02)`.

Agora, deve-se transformar as colunas em fatores

```
ex02$A<-as.factor(ex02$A)
ex02$B<-as.factor(ex02$B)
ex02$C<-as.factor(ex02$C)
ex02$bloco<-as.factor(ex02$bloco)
```

Com o comando `summary` obtém-se um resumo dos dados do objeto `ex02`.

```
> summary(ex02)
```

Explorando a variável resposta com o comando `tapply`

```
> ex02.mA <- tapply(resp, A, mean)
> ex02.mA

> ex02.mB <- tapply(resp, B, mean)
> ex02.mB

> ex02.mC <- tapply(resp, C, mean)
> ex02.mC
```

Análise de variância

A análise de variância para o experimento fatorial conduzido no delineamento em blocos completos casualizados pode ser obtida com o comando:

```
> ex02.av <- aov(resp ~ bloco + A + B + C + A*B + A*C + B*C + A*B*C)
```

ou

```
> ex02.av <- aov(resp ~ bloco + A*B*C)
> summary(ex02.av)
```

O uso da função `model.tables` permite a obtenção de estatísticas para cada um dos fatores do modelo e a combinação entre eles.

```
> ex02.mt <- model.tables(ex02.av, ty="means")
> ex02.mt
```

Análise de resíduos

Uma forma de analisar os resíduos é através do comando `plot(ex02.av)`. Esta função realiza a análise de resíduos do objeto `ex02.av`.

```
>par(mfrow=c(2,2))
>plot(ex02.av)
```

Uma outra maneira de realizar a análise de resíduos é gerando os gráficos individualmente. Desse modo existe uma maior liberdade para

```
> par(mfrow=c(1,1))
>  residuos <- resid(ex02.av)

>  plot.default(ex02$A, residuos)
>  title("Resíduos vs A")

>  plot.default(ex02$B, residuos)
>  title("Resíduos vs B")

>  plot.default(ex02$C, residuos)
>  title("Resíduos vs C")

> par(mfrow=c(2,2))
> preditos <- (ex02.av$fitted.values)
> plot(residuos, preditos)
> title("Resíduos vs Preditos")

s2 <- sum(resid(ex02.av)^2)/ex02.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")
```

O teste de Shapiro-Wilk também pode ser utilizado para avaliar a normalidade dos resíduos.

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


Análise da Interação

Inicialmente, faz-se o gráfico da interação para visualizar-se o comportamento entre as combinações dos diferentes níveis dos diferentes fatores.

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

interaction.plot(ex02$A, ex02$B, resp)
title("Interação A vs B")
interaction.plot(ex02$A, ex02$C, resp)
title("Interação A vs C")
interaction.plot(ex02$B, ex02$C, resp)
title("Interação P vs C")
```

Desdobramento da interação AC:

Estudaremos os níveis de A dentro de C.

Inicialmente, deve-se utilizar o modelo com o efeito de A aninhado em C.

```
ex02.avA<-aov(resp~bloc+C/A+A*B*C-A,data=ex02)
```

Observe que apenas nos interessa interpretar os efeitos do desdobramento.

```
summary(ex02.avA,split=list("C:A"=list(A.d.C0=1, A.d.C1=2)))
```

Para estudar os níveis de C dentro de A, basta considerar A/C .

Aula 1.1: Experimentos em esquema fatorial - com interação significativa

Objetivo dessa aula

O objetivo dessa aula é realizar a análise de variância de experimentos fatoriais. Será utilizado o Software R.

Procure entender os comandos utilizados e também interpretar os resultados.

Trabalhando com o arquivo de dados

Nesta aula iremos detalhar e discutir a análise do experimento fatorial 2×3 .

Este experimento, descrito na apostila do curso, comparou a resposta (altura) de mudas a diferentes recipientes e espécies de eucalipto.

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
```

Lembre-se de fornecer o caminho onde você salvou os dados.

Antes de começar as análises 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 para 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.

Análise descritiva

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 de interação.

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

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 você 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)
```

Análise de variância

Seguindo o modelo adequado, a análise de variância para esse 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

```

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.

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:

```

> par(mfrow=c(2,2))
> plot(ex04.av)

```

Os gráficos permitem uma análise dos resíduos que auxiliam no julgamento da adequabilidade do modelo. Evidentemente você não precisa se limitar aos 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
```

2. Desdobrando interações

Conforme 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
```

Teste de Tukey para comparações múltiplas

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

```

Exercícios

1. Experimento do Girocóptero: fatorial 2^3 completo. Considere o experimento com os

seguintes fatores e níveis:

- Asa: dobrada e não dobrada;
- Clip: um e dois clips;
- Cauda: uma e duas dobras.

Realize o experimento na sala de aula, considerando 4 repetições no delineamento em blocos completos casualizados . Não esqueça de aleatorizar os lançamentos.

Aula 2: Experimentos Fatoriais com Confundimento

Objetivo da aula

O objetivo desta aula é realizar a análise de experimentos fatoriais com confundimento.

Trabalhando com o arquivo de dados

Inicialmente, vamos trabalhar com o exemplo do experimento 3^3 com 9 blocos (bloquinhos), com uma repetição.

Observe que a forma de construção do arquivo de dados para análise no Ré realizado da mesma maneira que experimentos fatoriais. Basta fornecer as "coordenadas" para cada uma das respostas.

O arquivo pode ser obtido aqui. Se preferir, os dados podem ser inseridos no R através dos seguintes comandos:

```
A<-rep(rep(c(0,1,2),each=3),3)
B<-rep(rep(c(0,1,2),3),3)
C<-c(0,2,1,1,0,2,2,1,0,1,0,2,2,1,0,0,2,1,2,1,0,0,2,1,1,0,2)
bloco<-rep(c(0,1,2),each=9)
```

A variável resposta pode ser inserida com o comando `scan`:

```
prod<-scan() # pressione enter e inicie a digitação
```

Para gerar o data frame necessário para a análise, pode-se montar as colunas da seguinte forma:

```
ABC<-data.frame(bloco,A,B,C,prod)
```

Lembre-se que, para realizar a ANOVA, as colunas referentes aos fatores não podem ser numéricas, por isso, é necessário utilizar o comando `as.factor`:

```
A<-as.factor(A)
B<-as.factor(B)
C<-as.factor(C)
bloco<-as.factor(bloco)
```

Anova

Para fazer a análise de variância do experimento fatorial conduzido no esquema de confundimento, basta fornecer o modelo correspondente aos fatores que fazem parte do modelo.

Observe que nesse experimento, a interação tripla é quem foi confundida com blocos e existe apenas uma repetição do experimento. Por esse motivo a interação tripla não aparece no modelo. Nesse caso, os efeitos da interação tripla e do resíduo se confundem.

```
abc.av<-aov(prod~bloco+A+B+C+A*B+A*C+B*C,data=ABC)
summary(abc.av)
```

Verificando os pressupostos

Mesmo com apenas uma repetição, os resíduos ainda são estimados pelo modelo. Nesse caso, ainda podemos verificar os pressupostos da ANOVA. Discuta sobre isso!

```
par(mfrow=c(2,2))
plot(ex04.av)
```

Lembre-se que estes gráficos podem ser melhorados (personalizados).

Usando o pacote *conf.design*

Este pacote pode gerar o "confundimento" de fatores em experimentos fatoriais.

Use

```
require(conf.design)
```

para utilizar o pacote *conf.design*. Nesse pacote, será utilizada a função `conf.design` para gerar o confundimento.

```
conf.design(G, p, block.name="bloco", treatment.names=LETTERS[1:3])
```

G é uma matriz, que representa quais efeitos serão confundidos no experimento.

Por exemplo, no confundimento do fatorial 2^3 , a matriz G seria

```
g<-rbind(c(1,1,1))
```

ou seja, a interação tripla será confundida.

p é o número de níveis dos fatores.

block.name é o nome que será dado para os blocos, em geral, bloco.

treatment.names possibilita fornecer os nomes dos fatores. A função LETTERS coloca letras do alfabeto. De outro modo, pode-se inserir nomes em função do fenômeno.

Abaixo, são fornecidos alguns exemplos. Compare com os experimentos fornecidos no capítulo sobre experimentos com confundimento.

Fatorial 2^3

```
g<-rbind(c(1,1,1))
ABC.conf<-conf.design(g,2,block.name="Bloquinho",treatment.name=LETTERS[1:3])
ABC.conf
```

Fatorial 3^2

```
g<-rbind(c(1,1))
AB.conf<-conf.design(g,3,block.name="Bloquinho",treatment.name=LETTERS[1:2])
AB.conf
\begin{verbatim}
```

Fatorial 3^3

```
g<-rbind(c(1,1,1))
ABC.conf<-conf.design(g,3,block.name="Bloquinho",treatment.name=LETTERS[1:3])
ABC.conf
```

Para gerar repetições, use a função rep, e para montar o arquivo use scan() e data.frame

```
Bloquinho<-rep(ABC.conf$Bloquinho,4)
A<-rep(ABC.conf$A,4)
B<-rep(ABC.conf$B,4)
C<-rep(ABC.conf$C,4)
resp<-scan() # digite os dados
ABC.conf[,5]<-resp
ABC.conf<-data.frame(Bloquinho,A,B,C)
names(ABC.conf)[5]<-"resp" #coloque o nome na coluna
```

Exercícios

1. Considere o arranjo do experimento fatorial 2^3 em dois blocos (com confundimento), apresentado em sala de aula. Utilize a variável resposta do exemplo $2 \times 2 \times 2$ fornecido em sala. Analise os dados e compare com o resultado da análise anterior.
2. Considere os dados da tabela 1 de um experimento conduzido no esquema de confundimento no arranjo fatorial 3^2 com 3 repetições. Observe, pelos graus de liberdade da interação que o confundimento é parcial. Analise os dados desse experimento.

Tabela 1: Experimento fatorial 3^2 com confundimento parcial e 3 repetições.

Tratamentos	Blocos ou bloquinhos								
	I	II	III	IV	V	VI	VII	VIII	IX
A_0B_0	3196			3091			2781		
A_1B_1	3277			3363			3320		
A_2B_2	3222			3378			3423		
A_0B_2		2909			3322			3047	
A_1B_0		3287			3313			3332	
A_2B_1		2885			3731			2944	
A_0B_1			2530			3025			3286
A_1B_2			3233			3027			3519
A_2B_0			3679			3391			3639

3. Considere um experimento fatorial 2^3 com confundimento completo da interação tripla com 4 repetições. Baseado no quadro de sinais, foram formados os seguintes blocos:

Tabela 2: Experimento fatorial 2^3 com confundimento completo e 4 repetições.

Repetição I		Repetição II		Repetição III		Repetição IV	
Bloco 1	Bloco II	Bloco 1	Bloco II	Bloco 1	Bloco II	Bloco 1	Bloco II
ab=2173	b=2300	a=2228	ac=3495	(1)=2382	c=3528	abc=2996	ab=2393
(1)=2249	c=3538	abc=3592	(1)=2094	ab=2240	b=2118	c=1934	bc=2814
ac=3532	abc=3524	c=3116	bc=2249	bc=3495	abc=3350	b=2234	ac=3400
bc=2948	a=2319	b=2386	ab=2373	ac=2995	a=2272	a=2215	(1)=2297

Faça a análise de variância desse experimento.

4. Experimento do Girocóptero: fatorial 2^3 com confundimento completo.

Considere o experimento com os seguintes fatores e níveis:

- Asa: dobrada e não dobrada;
- Clip: um e dois clips;
- Cauda: uma e duas dobras.

Realize o experimento na sala de aula, considerando 4 repetições no delineamento em blocos completos casualizados (2 bloquinhos). Não esqueça de aleatorizar os lançamentos.

Soluções

- Exercício 1 .
- Exercício 2 .

Tabela 3: Estimativas de efeitos de um experimento 2^4

Efeitos	Estimativas
A	-8,00
B	24,00
C	-2,25
D	-5,50
AB	1,00
AC	0,75
AD	0,00
BC	-1,25
BD	4,50
CD	-0,25
ABC	-0,75
ABD	0,50
ACD	-0,25
BCD	-0,75
ABCD	-0,25

Aula 3: Gráfico Normal de Probabilidade.

Objetivo da aula

O objetivo dessa aula é construir o gráfico normal de probabilidade par avaliar os efeitos dos fatores de um experimento fatorial.

Fazendo o Gráfico

Para construir o gráfico, inicialmente necessitamos das estimativas dos efeitos dos fatores do experimento.

Considere o exemplo fornecido nas notas de aula de um experimento fatorial 2^4 com uma repetição. Os efeitos são apresentados na tabela a seguir:

As probabilidades de cada um dos efeitos são obtidas por $P = 100 * (i - 0,5)/15$.

O gráfico é obtido quando os efeitos *vs* P são colocados em um gráfico de dispersão. Uma linha reta entre os valores dos efeitos ajuda a verificar quais são os pontos que podem ser considerados como significativos.

Essa linha reta é colocada subjetivamente sobre os pontos. Uma regra (Montgomery, 2001) é ligar os pontos nos quantis 25% e 75%.

Os efeitos são:

```
efeitos<-c(-8,24,-2.25,-5.5,1,0.75,0,-1.25,4.5,-0.25,-0.75,0.5,-0.25,-0.75,-0.25)
```

Ordenando

```
efeitos.o<-sort(efeitos)
fatores<-c("A","B","C","D","AB","AC","AD","BC","BD","CD","ABC","ABD","ACD","BCD","ABCD")
names(efeitos)<-fatores
```

Aplicando os percentuais acumulados para os efeitos tem-se os seguintes valores:

```
P<-c(3.3,10,16.7,23.3,30,36.7,43.3,50,56.7,63.3,70,76.7,83.3,90,96.7)
```

Esses valores podem ser obtidos através da seguinte função no R:

```
P<-function(efeitos.o){
p<-numeric(0)
n<-1:length(efeitos)
for (i in n)
{
p[i]<-100*(i-.5)/length(efeitos)
}
print(p)
}
```

O gráfico é obtido pelo comando plot

```
plot(efeitos.o,p)
```

Para fazer a linha, podemos encontrar os quantis dos efeitos,

```
> quantile(efeitos.o)
 0%    25%    50%    75%   100%
-8.000 -1.000 -0.250  0.625 24.000
```

Utilizando o comando `locator` pode-se fazer uma linha entre os pontos dos quantis 25% e 75%. Depois do comando, clique nos pontos correspondentes no gráfico e uma linha será criada.

```
points(quantile(efeitos.o,.25),25,col=2,pch=19,cex=.5)
points(quantile(efeitos.o,.75), 75,col=2,pch=19,cex=.5)
locator(n=2,type="l") #uma linha entre dois pontos ou
segments(quantile(efeitos.o,.25),25, quantile(efeitos.o,.75), 75, col= 'blue')
```

Os efeitos significativos podem ser identificados com o comando `identify`. Lembre-se que os efeitos estão ordenados do menor para o maior.

```
identify(efeitos.o,p) # clique nos pontos do gráfico
```

Para terminar, aperte o botão direito do mouse (Linux). No Windows, aperte o botão direito e selecione `stop`.

Faça agora o Gráfico Normal de Probabilidade gerado pelo R. Compare os gráficos.

```
qqnorm(efeitos)
qqline(efeitos)
```

Experimente a opção

```
c<-qqnorm(efeitos,datax=T)
qqline(efeitos)
identify(c$x,c$y,labels=names(c$x))
```

E agora?

Analizando um experimento

Considere o seguinte experimento (Montgomery, 2001, exemplo 6.2) onde um experimento 2^4 foi realizado.

Utilize o Gráfico Normal de Probabilidade para avaliar a significância dos efeitos.

Inicialmente geramos a matriz com os sinais,

```
A<-rep(c(-1,1),8);A
B<-rep(c(-1,1),each=2,4);B
C<-rep(c(-1,1),each=4,2);C
D<-rep(c(-1,1),each=8);D
resp<-c(45,71,48,65,68,60,80,65,43,100,45,104,75,86,70,96)
```

Transforma-se os vetores em fatores

```
A<-as.factor(A)
B<-as.factor(B)
C<-as.factor(C)
D<-as.factor(D)
```

A análise de variância pode então ser realizada. Veja que, como não há repetições nesse caso e não há uma estrutura de confundimento, não são realizados testes na ANOVA.

```
abcd<-data.frame(A,B,C,D,resp)
abcd.av<-aov(resp~A*B*C*D,data=abcd)
summary(abcd.av)
efeitos<-abcd.av$eff[2:16]/2 #dois níveis
efeitos
```

Por isso, uma maneira de verificar a significância é utilizando o Gráfico Normal de Probabilidade. Identifique os pontos, utilizando o comando `identify`.

```
c<-qqnorm(efeitos,datax=T)
qqline(efeitos,datax=T)
identify(c$x,c$y,labels=names(c$x))
```

Identifique os pontos, utilizando o comando `identify`.

Exercícios

1. Analise os dados dos exercícios sobre confundimento utilizando o gráfico normal de probabilidade. Compare as conclusões com a análise anterior (**veja soluções**);
2. Escolha algum conjunto de dados analisado durante as aulas e tente aplicar o Gráfico Normal de Probabilidade. Compare as conclusões com a análise anterior;

Exercícios opcionais

1. Como exercício, estude e tente implementar o **Half Normal Plot** (Montgomery, 2001);
2. Como exercício, estude e tente implementar o Gráfico Normal de Probabilidade utilizando o *score z* (Montgomery, 2001).

Aula 4: Experimento Fatorial Fracionário: Fração meia

Objetivo da aula

O objetivo dessa aula é analisar dados de um experimento fatorial fracionário (fração meia). Inicialmente será realizada a análise de um experimento fatorial completo e depois a análise de um fração meia.

Nessa análise, será utilizado o conjunto de dados de Box, Hunter & Hunter(1978), sobre um reator com cinco fatores.

Organizando os dados

Os dados de um experimento fatorial completo podem ser arranjados da seguinte maneira:

A variável resposta:

```
y<-c(61,53,63,61,53,56,54,61,69,61,94,93,66,60,95,98,56,63,70,65,59,55,67,65,44,45,78,77,42,81,82)
```

E os fatores:

```
a<-rep(c(-1,1),16)
b<-rep(c(-1,-1,1,1),8)
c<-rep(c(-1,-1,-1,-1,1,1,1,1),4)
d<-rep(c(-1,1),each=8,2)
e<-rep(c(-1,1),each=16)
reactor1<-data.frame(a,b,c,d,e,y)
reactor1
```

Como há apenas dois níveis, não precisamos transformar para fatores.

Uma maneira alternativa é

```
require(AlgDesign)
x<-gen.factorial(2,5) #número de níveis, número de fatores
reactor2<-data.frame(x,y)
reactor2
```

Os dados também podem ser encontrados em

```
require(BsMD)
data(Reactor.data,package="BsMD")
print(Reactor.data)
```

Analizando os dados

Para construir o gráfico normal de probabilidade, inicialmente necessitamos das estimativas dos efeitos dos fatores do experimento. Por isso, deve-se fazer a análise de variância e extrair os efeitos. Observe que isso pode ser feito em qualquer dos conjuntos formados anteriormente.

```
r.av<-aov(y~a*b*c*d*e,data=reactor1)
summary(r.av)
r.av<-aov(y~X1*X2*X3*X4*X5,data=reactor2)
summary(r.av)
r.av<-aov(y~A*B*C*D*E,data=Reactor.data)
summary(r.av)
```

Separando os efeitos e construindo o gráfico normal de probabilidade:

```
efeitos<-coef(r.av)[2:32]*2 # 2 níveis
c<-qqnorm(efeitos)
qqline(efeitos)
identify(c$x,c$y,labels=names(c$y))
```

Observe nesse exemplo, que os efeitos de **B**, **D**, **E**, **BD** e **DE** são significativos.

O experimento fracionário

Utilizaremos as informações do experimento fatorial completo anterior para realizarmos a análise de um experimento fatorial fracionário. Ou seja, está sendo considerado que o experimento será iniciado a partir de agora. Isso nos fornece uma vantagem: pode-se comparar o resultado do experimento completo com o fracionário.

No R existem pelo menos duas maneiras de se obter uma configuração de um experimento fracionário. Uma é utilizando o pacote **AlgDesign** e a outra é utilizando o pacote **BHH2**.

Inicialmente, utilizaremos o pacote **AlgDesign**:

```
require(AlgDesign) # carregar o pacote
```

A função `gen.factorial()` é utilizada para gerar um experimento fatorial. Em seguida, cria-se uma quinta coluna para representar o fator E.

```
dat<-gen.factorial(levels=2,nVars=4,varNames=c("A","B","C","D")); dat
dat$E<-dat$A*dat$B*dat$C*dat$D; dat
```

Como estamos utilizando dados de um experimento fatorial completo, basta considerar as informações correspondentes para cada combinação dos níveis dos fatores e fazer a análise.

```
y<-c(56,53,63,65,53,55,67,61,69,45,78,93,49,60,95,82)
dados<-data.frame(dat,y);dados
colnames(dados)<-c(LETTERS[1:5], "y"); names(dados)
f.lm<-lm(y~A*B*C*D*E,data=dados)
```

Após a realização das análises, os efeitos podem ser avaliados com o uso do Gráfico Normal de Probabilidade.

```
efeitos<-f.lm$effects[2:16]
qqnorm(efeitos);qqline(efeitos)
identify(qqnorm(efeitos)$x,qqnorm(efeitos)$y)
efeitos
```

Em seguida, pode-se utilizar o pacote BHH2. Nesse pacote há uma função chamada `ffDesMatrix()` que faz as combinações para um experimento fracionário.

Primeiro, deve-se carregar o pacote:

```
require(BHH2)
```

Em seguida, utiliza-se a função `ffDesMatrix()`. Observe que a ordem das colunas do objeto será formada em função da forma como o gerador é inserido na função. Nesse exemplo, a coluna 5 será estruturada em função das colunas 1, 2, 3, 4 e 5 para construção de uma fração meia:

```
des<-ffDesMatrix(5,gen=list(c(5,1,2,3,4))) # 2^(5-1) fração meia
```

A entrada de dados e análise são como no exemplo anterior:

```
y<-c(56,53,63,65,53,55,67,61,69,45,78,93,49,60,95,82)
dados<-data.frame(des,y);dados

colnames(dados)<-c(LETTERS[1:5], "y"); names(dados)

f.lm<-lm(y~A*B*C*D*E,data=dados)
efeitos<-f.lm$effects[2:16]
qqnorm(efeitos);qqline(efeitos)
identify(qqnorm(efeitos)$x,qqnorm(efeitos)$y)
efeitos
```

Quais foram os efeitos significativos agora? E aí, vale a pena reduzir tanto o experimento?

O pacote FrF2

O pacote FrF2 pode ser utilizado para analisar os dados de um experimento fatorial. Além da avaliação dos efeitos do experimento, pode-se, por exemplo, utilizar a função `aliases()` para encontrar o padrão de confundimento do experimento e construir o cubo para representar um experimento fatorial.

Veja um exemplo com a utilização das análises do exemplo anterior:

```
require(FrF2)

summary(f.lm)
f.lm <- lm(y ~ (.)^2, data = dados) # apenas os efeitos estimáveis no modelo
summary(f.lm)

aliases(f.lm)
aliases(f.lm, code=TRUE) # padrão de confundimento

DanielPlot(f.lm)

MEPlot(f.lm) # efeitos no gráfico

IAPlot(f.lm) # gráfico de interação

cubePlot(f.lm, "A", "B", "C") # cubo
```

Exercícios

1. Faça uma nova configuração do fracionário, baseado nos critérios vistos em sala de aula e compare os resultados. Utilize os dados do exemplo dessa aula.
2. Planeje, execute e analise o experimento do Girocóptero, utilizando a fração meia, com a seguinte configuração:

Fator	Nome	-	+
A	Cauda	1 dobra	2 dobras
B	Clip	1 clip	2 clips
C	Asa	não dobrada	dobrada
D	Lançamento	reta	dobrada

Aula 5: Experimento Fatorial Fracionário: Fração um quarto (Girocóptero)

Objetivo da aula

O objetivo dessa aula é analisar dados de um experimento fatorial fracionário (fração um quarto).

Nessa análise, será utilizado o conjunto de dados gerado pelo experimento do Girocóptero com cinco fatores.

Realizando o experimento

O experimento será conduzido em sala de aula. Serão utilizados 5 fatores nesse experimento:

Tabela 4: Fatores e níveis para o experimento do Girocóptero.

Fator	Nome	-	+
A	Cauda	1 dobra	2 dobras
B	Clip	1 clip	2 clips
C	Asa	não dobrada	dobrada
D	Comp. Asa	inteira	cortada
E	Lançamento	reta	dobrada

Devem ser formados dois grupos para executarem os experimentos conforme o planejamento das tabelas 5 e 6.

Tabela 5: Grupo I: Experimento fatorial 2^{5-2} .

Ensaio	A	B	C	D=BC	E=ABC
1	-	-	-	+	-
2	+	-	-	+	+
3	-	+	-	-	+
4	+	+	-	-	-
5	-	-	+	-	+
6	+	-	+	-	-
7	-	+	+	+	-
8	+	+	+	+	+

Para analisar os dados, utilize os procedimentos da aula anterior.

Utilizando os pacotes BHH2 e FrF2 pode-se utilizar os seguintes procedimentos para gerar uma fração 1/4 de um experimento fatorial. Execute os comandos a seguir e interprete os

Tabela 6: Grupo II: Experimento fatorial 2^{5-2} .

Ensaio	A	B	C	D=-BC	E=-ABC
1	-	-	-	-	+
2	+	-	-	-	-
3	-	+	-	+	-
4	+	+	-	+	+
5	-	-	+	+	-
6	+	-	+	+	+
7	-	+	+	-	+
8	+	+	+	-	-

resultados:

```
require(BHH2)
require(FrF2)

des<-ffDesMatrix(6,gen=list(c(5,1,2,3),c(6,2,3,4))) # 2^(6-2)
des

y<-c(6,10,32,60,4,15,26,60,8,12,34,60,16,5,37,52)
dados<-data.frame(des,y);dados

colnames(dados)<-c(LETTERS[1:6],"y"); names(dados)

f.lm<-lm(y~A*B*C*D*E*F,data=dados)
efeitos<-f.lm$effects[2:16]
qqnorm(efeitos)
qqline(efeitos)

identify(qqnorm(efeitos)$x,qqnorm(efeitos)$y)
efeitos

aliases(f.lm)
DanielPlot(f.lm)
MEPlot(f.lm)
```

Aula 6: Superfície de Resposta

Objetivo da aula

O objetivo dessa aula é realizar uma análise de superfície de resposta utilizando o R.

O exemplo: tempo e temperatura

Utilizaremos os dados do exemplo estudado em sala de aula para exemplificar uma análise completa.

Inicialmente tem-se os dados referentes ao primeiro experimento, onde o pesquisador iniciou os estudos.

Lembrando que x_1 representa a variável tempo e x_2 a variável temperatura. A variável y representa o rendimento da reação em percentual.

Os dados do experimento podem ser utilizados da seguinte maneira:

```
x1<-c(-1,-1,1,1,0,0,0,0,0)
x2<-c(-1,1,-1,1,0,0,0,0,0)
y<-c(39.3,40,40.9,41.5,40.3,40.5,40.7,40.2,40.6)
dados1<-data.frame(x1,x2,y)
```

Em seguida faz-se uma análise de um modelo de regressão para encontrar a relação entre a variável resposta e as variáveis preditoras. Inicialmente, um modelo de primeira ordem é ajustado.

```
m1.1<-lm(y~x1+x2, data=dados1)
summary(m1.1) # estimativa dos efeitos
anova(m1.1)
```

Observe que nos resultados, os componentes inicialmente inseridos no modelo são significativos. Mas, é preciso ainda verificar se podem existir outros componentes como algum efeito de interação ou um algum efeito quadrático.

Call:

```
lm(formula = y ~ x1 + x2, data = dados1)
```

Residuals:

Min	1Q	Median	3Q	Max
-----	----	--------	----	-----

```
-0.244444 -0.044444 0.005556 0.055556 0.255556
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	40.44444	0.05729	705.987	5.45e-16 ***
x1	0.77500	0.08593	9.019	0.000104 ***
x2	0.32500	0.08593	3.782	0.009158 **

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

Residual standard error: 0.1719 on 6 degrees of freedom

Multiple R-Squared: 0.941, Adjusted R-squared: 0.9213

F-statistic: 47.82 on 2 and 6 DF, p-value: 0.0002057

O quadro da ANOVA pode ser obtido com o seguinte comando:

```
> anova(m1.1)
```

Observe que todos os efeitos são significativos.

Analysis of Variance Table

Response: y

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
x1	1	2.40250	2.40250	81.339	0.0001040 ***
x2	1	0.42250	0.42250	14.304	0.0091581 **
Residuals	6	0.17722	0.02954		

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

Ajuste do modelo

Após a análise de variância do modelo, é necessário verificar se o ajuste do modelo está adequado. Para isso, gera-se um modelo acrescentando-se o componente quadrático e a interação entre os fatores.

```
m2.1<-lm(y~x1+x2+x1*x2+I(x1^2), data=dados1)
anova(m2.1,test="F")
```

O resultado é

Analysis of Variance Table

Response: y

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
x1	1	2.40250	2.40250	55.8721	0.001713	**
x2	1	0.42250	0.42250	9.8256	0.035030	*
I(x1^2)	1	0.00272	0.00272	0.0633	0.813741	Quadrático
x1:x2	1	0.00250	0.00250	0.0581	0.821316	interação
Residuals	4	0.17200	0.04300			

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

Tanto o componente quadrático quanto o componente de interação foram não significativos ($p \leq 0,05$).

Maneira alternativa:

```
m1.lack<-lm(y~factor(x1)+factor(x2), data=dados1)
anova(m1.lack,test="F")
```

O resultado é o seguinte:

Analysis of Variance Table

Model 1: y ~ x1 + x2

Model 2: y ~ factor(x1) + factor(x2)

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	6	0.177222				
2	5	0.174500	1	0.002722	0.078	0.7912

Nesse caso, como o p-valor foi não significativo, considera-se que não há falta de ajuste desse modelo.

Ainda, uma outra maneira, é utilizar a função `pure.error.anova()` do pacote `alr3`.

Considerando o seguinte modelo:

```
m2.1<-lm(y~x1+x2+x1*x2, data=dados1)
```

Pode-se avaliar a falta de ajuste da seguinte maneira:

```
pure.error.anova(m2.1)
```

Analysis of Variance Table

Response: y

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
x1	1	2.40250	2.40250	55.8721	0.001713	**
x2	1	0.42250	0.42250	9.8256	0.035030	*
x1:x2	1	0.00250	0.00250	0.0581	0.821316	
Residuals	5	0.17472	0.03494			
Lack of fit	1	0.00272	0.00272	0.0633	0.813741	
Pure Error	4	0.17200	0.04300			

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

Nessa análise deve-se observar o p-valor dos desvios de regressão (Lack of fit). Veja que o p-valor foi não significativo, indicando que não há falta de ajuste nesse modelo.

Segunda Ordem

Tem-se agora, os dados do segundo experimento, onde o pesquisador tem o interesse de ajustar o modelo de segunda ordem:

```
x1<-c(-1,-1,1,1,0,0,0,0,0)
x2<-c(-1,1,-1,1,0,0,0,0,0)
y<-c(76.5,77,78,79.5,79.9,80.3,80,79.7,79.8)
dados2<-data.frame(x1,x2,y)
```

A análise do modelo inicial é feita da seguinte maneira

```
m2.1<-lm(y~x1+x2,data=dados2)
summary(m2.1)
anova(m2.1)
```

Ajuste do modelo

A análise do ajuste é dada da seguinte forma:

```
m2.2<-lm(y~x1+x2+x1*x2+I(x1^2), data=dados2)
anova(m2.2,test="F")
```

Observe que nesse caso, o componente quadrático foi significativo ($p < 0,05$).

Usando uma maneira alternativa:

```
m2.lack<-lm(y~factor(x1)+factor(x2), data=dados2)
anova(m2.1,m2.lack,test="F")
```

Construindo a Superfície de Resposta

Os dados do experimento aumentado são:

```
x1<-c(-1,-1,1,1,0,0,0,0,0,1.414,-1.414,0,0)
x2<-c(-1,1,-1,1,0,0,0,0,0,0,0,1.414,-1.414)
y<-c(76.5,77,78,79.5,79.9,80.3,80,79.7,79.8,78.4,75.6,78.5,77)
dados3<-data.frame(x1,x2,y)
```

Encontrando o Modelo

O modelo pode ser construído da seguinte maneira:

```
m1.3<-lm(y~x1*x2+I(x1^2)+I(x2^2),data=dados3)
summary(m1.3)
anova(m1.3)
```

Observe que os resíduos já são pequenos. O que você acha da falta de ajuste?

Construindo a Superfície de Resposta

Uma das maneiras é utilizando a função `persp`:

Inicialmente, deve ser gerada uma sequência de valores dos níveis dos fatores.

```
x<-seq(-1.5,1.5,length=30)
y<-seq(-1.5,1.5,length=30)
```

```
z<-function(x=x,y=y){y<-m1.3$coef[1]+m1.3$coef[2]*x+m1.3$coef[3]*y+
m1.3$coef[4]*x^2+m1.3$coef[5]*y^2+m1.3$coef[6]*x*y}
#alternativamente, os próprios coeficientes podem ser digitados
#z<-function(x,y){y<-79.94+0.99*x+0.52*y-1.38*x^2-1*y^2+0.25*x*y}
```

O comando `outer` prepara os dados para serem utilizados pela função `persp`

```
z <- outer(x, y, z)
persp(x,y,z,theta = -35, phi = 5, expand = 0.5, col = "gray",xlab="Tempo",
ylab="Temperatura",zlab="Rend",scale=T,ticktype = "detailed")
```

Experimente alterar as opções gráficas!

Construindo as Curvas de Nível

Pode-se visualizar o comportamento da superfície através de curvas de nível do seguinte modo:

```
contour(x,y,z,xlab="Tempo",ylab="Temperatura")
```

Outras representações gráficas podem ser utilizadas.

```
image(x,y,z,col=topo.colors(12))
image(x,y,z,col=terrain.colors(12))
image(x,y,z,col=heat.colors(12))
```

Experimente também

```
image(x,y,z,col=topo.colors(12))
contour(x,y,z,xlab="Tempo",ylab="Temperatura",add=T)
```

Estudando o ponto estacionário

Utilizaremos os procedimentos propostos em sala

```
b<-matrix(c(m1.3$coef[2],m1.3$coef[3]));b
B<-matrix(c(m1.3$coef[4],m1.3$coef[6]/2,m1.3$coef[6]/2,m1.3$coef[5]),ncol=2);B

x0<--0.5*solve(B)%*%b ; x0 # coordenadas do ponto máximo

# correspondente em tempo e temperatura (var naturais)

tempo<-x0[1]*5+85 ;tempo
temperatura<-x0[2]*5+175 ; temperatura
```

Experimente observar o seguinte gráfico:


```
image(x,y,z,col=topo.colors(12))
contour(x,y,z,xlab="Tempo",ylab="Temperatura",add=T)
points(x0[1],x0[2],col="red")
```

O ponto "máximo" é dado por

```
y0<-m1.3$coef[1]+1/2*t(x0)%*%b
```

Análise canônica: estudando a forma da superfície

Para analisar o ponto estacionário devemos observar os sinais das raízes.

```
raizes<-eigen(B)

> raizes
$values
[1] -0.9634986 -1.4142867

$vectors
      [,1]      [,2]
[1,] 0.2897174 0.9571122
[2,] 0.9571122 -0.2897174
```

Como os sinais são negativos, o ponto é de máximo.