

Aulas Práticas
CE213: Planejamento de Experimentos I

Adilson dos Anjos

Última atualização: 17 de abril de 2006

Aula 01: Execução e análise de uma experimento

Objetivo desta aula

O objetivo desta aula é apresentar um experimento com problemas de planejamento, analisar estes problemas e implementar um experimento sem problemas de planejamento.

Será utilizado nesta aula, um programa desenvolvido através da linguagem Java.

Esta atividade foi desenvolvida por C. M. Anderson-Cook e Sundar Dorai-Raj e publicada no Journal of Statistics Education. O programa pode ser obtido na página on o trabalho foi publicado.

Observe no artigo que existem algumas referências para outros sites que também utilizam procedimentos semelhantes para o ensino de métodos estatísticos. Vale a pena!

Para o programa Java funcionar corretamente é necessário que o seu browser esteja habilitado para funcionar com a linguagem Java. Os arquivos não necessitam de instalação. Basta copiá-los para o computador e abrir o arquivo `index.html`.

Existe a possibilidade de utilizar o aplicativo diretamente pela internet sem a necessidade de copiar os arquivos.

Abaixo, são apresentados alguns links onde aplicativos Java são utilizados em procedimentos estatísticos:

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

Aula 02: Delineamento Completamente Casualizado

Objetivo desta Aula

O objetivo desta aula é utilizar o software R para realizar a análise de variância de um experimento conduzido no Delineamento Completamente Casualizado.

Trabalhando com o arquivo de dados

A seguir são apresentados os comandos para a análise do experimento. Procure entender o que cada comando executa e compare as saídas com os resultados apresentados em sala de aula.

Inicialmente, o arquivo de dados está disponível em arquivo de dados que deve ser 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)
```

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

Nota: na ANOVA, as variáveis independentes precisam possuir a característica "factor". Caso contrário, o R realizará uma análise de regressão entre as variáveis.

Análise descritiva

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)
```

Pode-se “desanexar” o objeto com os dados (embora isto não seja obrigatório) com o comando.

```
detach(ex01)
```

Quando um objeto do tipo *list* ou *data.frame* é anexado no caminho de procura com o comando `attach()` faz-se com que os componentes deste objeto se tornem imediatamente disponíveis e portanto pode-se, por exemplo, digitar somente `trat` ao invés de `ex01$trat`.

Prosseguindo 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)
```

ANOVA

O comando `aov`, realiza a análise dos dados do `data.frame`. Compare a saída do comando `aov` com o comando `ANOVA`.

```
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 de variância e outras informações.

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

```
ex01.av$res
residuals(ex01.av)
```

Após realizar a ANOVA, é necessário verificar os pressupostos do modelo:

Análise de resíduos

Homocedasticidade:

Graficamente, podemos analisar a homocedasticidade através de um box-plot,

```
boxplot(ex01.av$res ~ trat,ylab="Resíduos",xlab="Linhagens")
```

Através de um gráfico dos resíduos vs tratamentos,

```
plot.default(trat,ex01.av$res,ylab="Resíduos",xlab="Linhagens")
```

Ainda, pode-se avaliar através de um teste, como por exemplo o teste de Bartlett:

```
bartlett.test(ex01.av$res, trat)
```

Normalidade:

Graficamente, pode-se avaliar a normalidade dos resíduos fazendo

```
hist(ex01.av$res, main=NULL)
title("Histograma dos Resíduos")
```

```
stem(ex01.av$res)
```

```
qqnorm(ex01.av$res,ylab="Resíduos", main=NULL)
qqline(ex01.av$res)
title("Grafico Normal de Probabilidade dos Resíduos")
```

Através do teste de Shapiro-Wilk:

Teste de Shapiro-Wilk para Normalidade

Estatística do Teste

O objetivo deste teste é fornecer uma estatística de teste para avaliar se uma amostra tem distribuição Normal. O teste pode ser utilizado para amostras de qualquer tamanho.

A estatística \mathbf{W} de teste para normalidade é definida como

$$W = \frac{b^2}{s^2} = \frac{(\sum_{i=1}^n a_i y_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (1)$$

onde

y_i é a variável aleatória observada e a_i são coeficientes tabelados.

Execução do teste:

Par calcular a estatística \mathbf{W} , de uma mostra aleatória de tamanho n , dada por y_1, y_2, \dots, y_n , procede-se da seguinte forma:

1. Ordenar as observações em ordem decrescente: $y_1 \leq y_2 \leq \dots \leq y_n$.
2. Calcular s^2

$$s^2 = \sum_{i=1}^n (y_i - \bar{y})^2$$

3. (a) Se n é par, $n = 2k$, faz-se

$$b = \sum_{i=1}^k a_{n-i+1} (y_{n-i+1} - y_i)$$

os valores de a_{n-i+1} são tabelados.

3. (b) Se n é ímpar, $n = 2k + 1$, os cálculos permanecem os mesmos, exceto que, $a_{k+1} = 0$

$$b = a_n (y_n - y_1) + \dots + a_{k+2} (y_{k+2} - y_k)$$

4. Calcular

$$W = \frac{b^2}{s^2}$$

5. Avaliar a estatística do teste através do P-valor. No caso de uma valor significativo para a estatística do teste, isso indica falta de normalidade para a variável aleatória analisada.

```
shapiro.test(ex01.av$res) #teste para normalidade
```

Independência:

A independência, com algumas restrições, pode ser analisada graficamente, através de

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

Ainda é possível avaliar algum tipo de dependência através da ordenação dos resíduos, caso exista uma ordem de obtenção dos dados conhecida:

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

Verificação de Outliers:

Utilizando o critério de +3 ou -3 desvios padronizados, pode-se avaliar a existência de candidatos à outlier utilizando os seguintes comandos:

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

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

```
plot(ex01.av)
```

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

```
names(anova(ex01.av))
```

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

```
res <- ex01.av$res # extraíndo resíduos
```

```
respad <- (res/sqrt(s2)) # resíduos padronizados
```

```
boxplot(respad)
```

```
title("Resíduos Padronizados" )
```

```
plot.default(ex01$strat, respad, xlab="Linhagens")
```

```
title("Resíduos Padronizados" )
```

Teste de Tukey para comparações múltiplas

No R, o teste de Tukey é apresentado através de intervalos de confiança. A interpretação é: se o intervalo de confiança para a diferença entre duas médias não incluir o valor zero, significa que rejeita-se a hipótese nula, caso contrário, não rejeita-se.

O resultado pode ser visto através de uma tabela e/ou graficamente:

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

Uma outra maneira é utilizar a seguinte função para estes dados.

```
## Diferença entre médias para um fator e igual número de repetições
#r=número de repetições
#t=número de tratamentos

dif.medias<-function(dados=ex01,r=6, t=9,alpha=0.95 )
{ attach(dados)
  modelo<-aov(resp~trat,data=dados)
  trat.m<-tapply(resp,trat,mean)
  trat.m1<-trat.m
  m1d<-outer(trat.m1,trat.m1,"-")
  m1d<-m1d[lower.tri(m1d)]
  m1n<-outer(names(trat.m1),names(trat.m1),paste,sep="-")
  names(m1d)<-m1n[lower.tri(m1n)]
  s2<-sum(resid(modelo)^2)/modelo$df.res
  n<-r
  dif.t<-qtukey(alpha,t,modelo$df.res)*sqrt(s2/n)
  data.frame(dif=m1d,sig=ifelse(abs(m1d)>dif.t,"*", "ns"))
}
```

Contrastes

Em alguns casos, pode-se ter interesse em estudar contrastes específicos. No exemplo das linhagens, temos 9 tratamentos que possibilitam construir 8 contrastes ortogonais e independentes. Os contrastes poderiam ser:

$$\begin{aligned} \hat{Y}_1 &= 1/4L_1 + 1/4L_2 + 1/4L_3 + 1/4L_4 - 1/5L_5 - 1/5L_6 - 1/5L_7/1/5L_8 - 1/5L_9 \\ \hat{Y}_2 &= 1L_1 + 1L_2 - 1L_3 - 1L_4 \\ \hat{Y}_3 &= 1L_1 - 1L_2 \\ \hat{Y}_4 &= 1L_3 - 1L_4 \\ \hat{Y}_5 &= 1L_5 + 1L_6 + 1L_7 + 1L_8 - 4L_9 \\ \hat{Y}_6 &= 1L_5 + 1L_6 - 1L_7 - 1L_8 \\ \hat{Y}_7 &= 1L_5 - 1L_6 \\ \hat{Y}_8 &= 1L_7 - 1L_8 \end{aligned}$$


```
> ex01.av$cont
$trat
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
t1  0.25   1   1   0   0   0   0   0
t2  0.25   1  -1   0   0   0   0   0
t3  0.25  -1   0   1   0   0   0   0
t4  0.25  -1   0  -1   0   0   0   0
t5 -0.20   0   0   0   1   1   1   0
t6 -0.20   0   0   0   1   1  -1   0
t7 -0.20   0   0   0   1  -1   0   1
t8 -0.20   0   0   0   1  -1   0  -1
t9 -0.20   0   0   0  -4   0   0   0
```

Exercícios

1. Analise o experimento da página 63 das notas de aula da disciplina de Planejamento de Experimentos I.
Clique aqui para ver e copiar o arquivo de dados.
2. Analise o experimento da página 64 das notas de aula da disciplina de Planejamento de Experimentos I.
Clique aqui para ver e copiar o arquivo de dados.

Opcional:

Nesta seção você poderá realizar a ANOVA usando operações aritméticas no R.

Utilize o arquivo de dados sobre linhagens como exemplo.

```
n <- length(resp)
n
nt <- length(levels(trat))
nt

correcao <- ((sum(resp))^2)/n
correcao

gltot <- n - 1
gltra <- nt - 1
glres <- n - nt
```

```
sqtot <- sum(resp^2) - correcao
sqtot

trtot <- tapply(resp, trat, sum)
sqtra <- (sum((trtot)^2))/6 - correcao
sqtra

sqres <- sqtot - sqtra
sqres

qmtra <- sqtra/gltra
qmres <- sqres/glres

fval <- qmtra/qmres
fval
pval <- pf(fval, gltra, glres, lower.tail = F)
pval

## Definindo o quadro da ANOVA

qav <- matrix(NA, nr=3, nc=5)
dimnames(qav) <- list(c("Tratamentos", "Residuo", "Total"),
c("GL", "SQ", "QM", "valorF", "valorP"))
qav

## Montando o Quadro da ANOVA

qav[1,] <- c(gltra, glres, qmtra, fval, pval)
qav[2,1:3] <- c(sqtra, sqres, qmres)
qav[3,1:2] <- c(gltot, sqtot)
qav
```

Simulação de um experimento

Uma indústria precisa decidir qual produto será utilizado para formulação de um detergente. Na composição, um ácido faz parte da fórmula. Esse ácido precisa ser dissolvido em água para ser misturado à fórmula do detergente. Quatro produtos estão sendo avaliados.

É importante nesse experimento que a dissolução do ácido seja rápida para que o processo

de fabricação seja eficiente.

Simulação:

Utilizando comprimidos efervescentes para simular os ácidos, será realizado um experimento.

Quatro tipos(marcas) de comprimidos serão utilizados.

Em cada experimento serão realizadas 3 repetições em um delineamento completamente casualizado.

Deverão ser formados dois grupos para realizarem o experimento. Cada grupo receberá uma amostra extra para realizar um experimento piloto.

A variável resposta de interesse é o tempo de dissolução do comprimido por completo.

Obtenha os dados e faça a análise estatística.

Bom trabalho!

Aula 03: Transformação de dados

Objetivos desta aula

Os objetivos desta aula são analisar um conjunto de dados que apresentam problemas com relação aos pressupostos e como resolver essa situação.

Trabalhando com o arquivo de dados

Tranformação de dados é uma das possíveis formas de contornar o problema de dados que não obedecem aos pressupostos da análise de variância. Isso poder ser feito com relativa facilidade no programa R.

Considere o seguinte exemplo da página 64 das notas de aula do curso. Os dados referem-se ao número de falhas eletrônicas em diferentes sistemas de atendimento telefônico no período de um ano de funcionamento (Delineamento completamente asualizado).

Uma maneira alternativa de entrar com os dados é utilizando a função `scan` e montar um data-frame.

```
y <- scan()
1: 2370
2: 1687
3: 2592
...
30: 44
31:
Read 30 items

tr <- data.frame(trat = rep(1:5, each=6), resp = y)
tr
```

Lembre-se que *trat* precisa ser inserido como uma variável não numérica

```
tr$trat<-as.factor(tr$trat)
```

ANOVA e Pressupostos

A seguir ajusta-se o modelo e inspecionam-se os resíduos.

```
##Anova

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

##Homocedasticidade

plot.default(tr$trat, tr.av$res)

##Normalidade

qqnorm(tr.av$res, ylab="Resíduos", main=NULL)
qqline(tr.av$res)
title("Grafico Normal de Probabilidade dos Resíduos")
```

O gráfico de resíduos vs valores preditos mostra claramente uma heterogeneidade de variâncias e o $QQ - plot$ mostra um comportamento dos dados que se afasta muito da distribuição normal. A mensagem é clara mas testes podem ser feitos para verificar o desvio dos pressupostos.

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

      Bartlett test for homogeneity of variances

data:  tr$resp and tr$trat
Bartlett's K-squared = 29.586, df = 4, p-value = 5.942e-06

shapiro.test(tr.av$res)

      Shapiro-Wilk normality test

data:  tr.av$res
W = 0.8961, p-value = 0.006742
```

Nos resultados acima a homogeneidade de variâncias foi rejeitada e também a normalidade dos resíduos.

Definindo a transformação

Para tentar contornar o problema utiliza-se a transformação Box-Cox, que consiste em transformar os dados de acordo com a expressão

$$Y^* = Y^\lambda$$

onde λ é um parâmetro a ser estimado dos dados. Se $\lambda = 0$ a equação acima se reduz a

$$Y^* = \log(Y)$$

onde \log é o logaritmo neperiano. Uma vez obtido o valor de λ encontra-se os valores dos dados transformados conforme a equação acima e utiliza-se estes dados transformados para efetuar as análises.

A função `boxcox` do pacote MASS calcula a verossimilhança perfilhada do parâmetro λ . Deve-se escolher o valor que maximiza esta função. Nos comandos a seguir inicia-se carregando o pacote MASS e depois obtém-se o gráfico da verossimilhança perfilhada. Neste caso, é de interesse obter o valor máximo de λ . Um gráfico com um zoom na região de interesse fornece esse valor.

```
require(MASS)
box.tr<-boxcox(resp ~ trat, data=tr, plotit=T)
box.tr<-boxcox(resp ~ trat, data=tr, lam=seq(-1, 1, 1/10))
```

O gráfico mostra que o valor que maximiza a função é aproximadamente $\hat{\lambda} = 0.19$. Abaixo, uma forma de se obter o valor exato de $\hat{\lambda}$

Transformação BOX-COX

```
lambda <- box.tr$x[which(box.tr$y == max(box.tr$y))]

lambda
[1] 0.1919192
```

Desta forma o próximo passo é obter os dados transformados e depois realizar as análises utilizando estes novos dados.

```
tr$respt <- tr$resp^(lambda)
tr.avt <- aov(respt ~ trat, data=tr)
```

Note que os resíduos tem um comportamento bem melhor do que o observado para os dados originais. A análise deve prosseguir utilizando-se então os dados transformados.

```
bartlett.test(tr$respt, tr$trat)
shapiro.test(tr.avt$res)
```

NOTA: No gráfico da verossimilhança perfilhada notamos que é mostrado um intervalo de confiança para λ e que o valor 0 está contido neste intervalo. Isto indica que podemos utilizar a transformação logarítmica dos dados e os resultados serão bem próximos dos obtidos com a transformação previamente adotada.

```
tr.avl <- aov(log(resp) ~ trat, data=tr)
plot(tr.avl)
```

Exercícios

1. Analise o experimento da página 64, tabela 11, das notas de aula da disciplina de Planejamento de Experimentos I. Veja que os dados deste experimento possuem problemas com relação aos pressupostos. Descubra qual transformação pode resolver este problema! Clique aqui para ver e copiar o arquivo de dados.
2. Compare o resultado do teste de comparações múltiplas (Tukey) com os dados não transformados e transformados. As conclusões foram as mesmas?

Aula 04: Delineamento em Blocos Completos Casualizados

Objetivos da Aula

O objetivo desta aula é utilizar o software R para realizar a análise de variância de um experimento conduzido no Delineamento em Blocos Completos Casualizados.

Trabalhando com o arquivo de dados

Serão utilizados os dados do experimento sobre percentual de óleo em *S. linicola* em diferentes estágios de crescimento, conduzido no Delineamento em Blocos Completos Casualizados.

A seguir são apresentados os comandos para a análise do experimento. Procure entender o que cada comando executa e compare as saídas com os resultados apresentados em sala de aula.

Inicialmente, o arquivo de dados está disponível em arquivo de dados que deve ser copiado para o seu diretório de trabalho.

Inicialmente vamos entrar com os dados no R. Há várias possíveis maneiras de fazer isto. Vamos aqui usar a função *scan* e entrar com os dados por linha da tabela. Digitamos o comando abaixo e a função *scan* recebe os dados. Depois de digitar o último dado digitamos ENTER em um campo em branco e a função encerra a entrada de dados retornando para o prompt do programa.

OBS: Note que, sendo um programa escrito na língua inglesa, os decimais devem ser indicados por '.' e não por vírgulas.

```
> y <- scan()
1: 4.4
2: 5.9
3: 6.0
...
24: 6.7
25:
Read 24 items
```

Agora vamos montar um *data.frame* com os dados e os indicadores de blocos e tratamentos.

```
bc01 <- data.frame(estag = factor(rep(1:6, each=4)), bloco=factor(rep(1:4, 6)), resp=y)
```

Note que usamos a função `factor` para indicar que as variáveis `bloco` e `estag` são níveis de fatores e não valores numéricos.

Pode-se utilizar o comando `read.table` para ler o arquivo de dados:

```
bc01 <- read.table("exemplo04.txt", header=T)
bc01
```

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

Explorando os dados

Vamos agora explorar um pouco os dados.

```
names(bc01)
summary(bc01)

attach(bc01)

plot(resp ~ estag + bloco)

bc01.mt <- tapply(resp, estag, mean)
bc01.mt
bc01.mb <- tapply(resp, bloco, mean)
bc01.mb

plot.default(estag, resp)
points(bc01.mt, pch="x", col=2, cex=1.5)

plot.default(bloco, resp)
points(bc01.mb, pch="x", col=2, cex=1.5)
```

Nos gráficos e resultados acima procuramos captar os principais aspectos dos dados bem como verificar se não há interação entre blocos e tratamentos, o que não deve acontecer neste tipo de experimento.

A seguir vamos ajustar o modelo e obter outros resultados, incluindo a análise de resíduos e testes para verificar a validade dos pressupostos do modelo.

ANOVA

```
bc01.av <- aov(resp ~ bloco + estag)
anova(bc01.av)
```

```
names(bc01.av)
```

Análise de resíduos

Graficamente

```
par(mfrow=c(2,2))
plot(bc01.av)
```

Homocedasticidade, Normalidade e Independência

```
residuos <- (bc01.av$residuals)
```

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

```
plot(bc01$estag,residuos)
title("Resíduos vs Estágios \n Homocedasticidade")
```

```
preditos <- (bc01.av$fitted.values)
```

```
plot(residuos,preditos)
title("Resíduos vs Preditos \n Independência")
```

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

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

```
respad <- (residuos/sqrt(anova(bc01.av)$"Mean Sq"[2]))
boxplot(respad)
title("Resíduos Padronizados - outliers")
```

```
plot(bc01$bloco,residuos)
title("Resíduos vs Blocos")
```

Teste para Normalidade dos Resíduos

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

Como foi detectado efeito de tratamentos faz-se um teste de comparações múltiplas e encerra-se as análises desanexando o objeto do caminho de procura.

Teste para Comparações Múltiplas

```
bc01.tk <- TukeyHSD(bc01.av, "estag", ord=T)
bc01.tk
plot(bc01.tk)

detach(bc01)
```

Exercícios

1. Analise o experimento da página 72, tabela 13, das notas de aula da disciplina de Planejamento de Experimentos I.
Clique aqui para ver e copiar o arquivo de dados.
2. Analise o experimento da página 72, tabela 14, das notas de aula da disciplina de Planejamento de Experimentos I.
Clique aqui para ver e copiar o arquivo de dados.

Opcional:

Apesar de não ter sido estudado durante as aulas, uma das pressuposições do modelo proposto para a análise dos dados de um Delineamento em Blocos Completos Casualizados é que não ocorra interação entre blocos e tratamentos. Este conceito de interação será estudado na disciplina Planejamento de Experimentos II.

Em geral, por conveniência, esse pressuposto não é analisado. Quando há problemas desta natureza, pode haver problemas com outros pressupostos, uma vez que o modelo pode não estar sendo utilizado de forma adequada.

Mesmo assim, para os mais interessados, abaixo é apresentada a forma de verificação deste pressuposto da ANOVA.

Testando a não aditividade, primeiro extrai-se coeficientes de tratamentos e blocos

```
bc01.av$coeff
bl <- c(0, bc01.av$coeff[2:4])
tr <- c(0, bc01.av$coeff[5:9])
bl
tr
```

agora cria-se um novo termo e testa-se sua significância na ANOVA

```
bltr <- rep(bl, 6) * rep(tr, rep(4,6))

ttna <- update(bc01.av, .~. + bltr)
anova(ttna)
```

observe a significância do termo *bltr*.

Outra forma, é através de uma análise gráfica:

```
interaction.plot(estag, bloco, resp)
interaction.plot(bloco, estag, resp)
```

Os resultados acima indicam que os pressupostos estão obedecidos para este conjunto de dados e a análise de variância é válida.

Aula 05: Delineamento em Quadrado Latino

Objetivos da Aula

O objetivo desta aula é utilizar o software R para realizar a análise de variância de um experimento conduzido no Delineamento em Quadrado Latino.

Trabalhando com o arquivo de dados

Serão utilizados os dados do experimento sobre produção de grãos de diferentes variedades de feijão, página 76 das notas de aula.

A seguir são apresentados os comandos para a análise do experimento. Procure entender o que cada comando executa e compare as saídas com os resultados apresentados em sala de aula.

Inicialmente, o arquivo de dados está disponível em arquivo de dados que deve ser copiado para o seu diretório de trabalho.

Inicialmente vamos entrar com os dados no R. Há várias possíveis maneiras de fazer isto. Vamos aqui usar a função `scan` e entrar com os dados por linha da tabela. Digitamos o comando abaixo e a função `scan` recebe os dados. Depois de digitar o último dado digitamos ENTER em um campo em branco e a função encerra a entrada de dados retornando para o prompt do programa.

OBS: Note que, sendo um programa escrito na língua inglesa, os decimais devem ser indicados por '.' e não por vírgulas.

```
y <- scan()
1: 7.6
2: 8.2
3: 10.4
...
25: 7.5
26:
Read 25 items
```

Agora vamos montar um `data.frame` com os dados e os indicadores de tratamentos, linhas e colunas:

```
trat<-c("B","A","D","E","C","C","B","E","A","D","A","D","B","C","E","D","E","C","B","A","E","C","A","D","B")
trat<-as.factor(trat)

lat01 <- data.frame(linha=factor(rep(1:5, each=5)), coluna=factor(rep(1:5,5)), trat=trat, resp=y)
```

Note que usamos a função `factor` para indicar que as variáveis linhas e colunas são níveis de fatores e não valores numéricos.

O mesmo vale para tratamentos.

Pode-se utilizar o comando `read.table` para ler o arquivo de dados:

```
lat01 <- read.table("latino01.txt", header=T)
lat01
```

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

Explorando os dados

Vamos agora explorar um pouco os dados.

```
names(lat01)
summary(lat01)

attach(lat01)

plot(resp ~ coluna + linha + trat)

lat01.mt <- tapply(resp, trat, mean)
lat01.mt
lat01.ml <- tapply(resp, linha, mean)
lat01.ml
lat01.mc <- tapply(resp, coluna, mean)
lat01.mc

plot.default(trat, resp)
points(lat01.mt, pch="x", col=2, cex=1.5)
```

Nos gráficos e resultados acima procuramos captar os principais aspectos dos dados bem como verificar se não há interação entre linhas, colunas e tratamentos, o que não deve acontecer neste tipo de experimento.

A seguir vamos ajustar o modelo e obter outros resultados, incluindo a análise de resíduos e testes para verificar a validade dos pressupostos do modelo.

ANOVA

```
lat01.av <- aov(resp ~ coluna + linha + trat)
anova(lat01.av)
names(lat01.av)
```

Análise de resíduos

Graficamente

```
par(mfrow=c(2,2))
plot(lat01.av)
```

Homocedasticidade, Normalidade e Independência

```

residuos <- (lat01.av$residuals)

par(mfrow=c(2,2))

plot(lat01$trat,residuos)
title("Resíduos vs Estágios \n Homocedasticidade")

preditos <- (lat01.av$fitted.values)

plot(residuos,preditos)
title("Resíduos vs Preditos \n Independência")

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

par(mfrow=c(2,1))

respad <- (residuos/sqrt(anova(lat01.av)$"Mean Sq"[4]))
boxplot(respad)
title("Resíduos Padronizados - outliers")

outlier<-c(max(respad),min(respad))
outlier

```

Teste para Normalidade dos Resíduos

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

Como foi detectado efeito de tratamentos faz-se um teste de comparações múltiplas e encerra-se as análises desanexando o objeto do caminho de procura.

Teste para Comparações Múltiplas

```

lat01.tk <- TukeyHSD(lat01.av, "trat", ord=T)
lat01.tk
plot(lat01.tk)

detach(lat01)

```

Exercícios

Analise o experimento da página 80, tabela 16, das notas de aula da disciplina de Planejamento de Experimentos I.

Dado o seguinte conjunto de dados de um experimento conduzido no delineamento em quadrado latino, faça a análise de variância e verifique os pressupostos.

Tabela 1: Experimento em Quadrado Latino

Linhas	Colunas				
	I	II	III	IV	V
I	432 D	518 A	458 B	583 C	331 E
II	724 C	478 E	524 A	550 B	400 D
III	489 E	384 B	556 C	297 D	420 A
IV	494 B	500 D	313 E	486 A	501 C
V	515 A	660 C	438 D	394 E	318 B

Aula 06: Determinação do Número de Repetições

Objetivos da Aula

O objetivo desta aula é utilizar o software R para calcular o número de repetições de um experimento através do método de Tukey.

Definindo a expressão

A expressão para definição do número de repetições é definida por

$$r = \frac{q^2 s^2 F}{d^2}$$

Criando uma Função

No R(), pode-se criar uma função que realiza este cálculo para um experimento em bolcos completos casualizados:

```
function(n2,s,t,d,ri,alpha=0.95)
{
## Cálculo do número de repetições para um experimento em BCC
```

```
## n2 = graus de liberdade do resíduo de um experimento anterior
## s = desvio padrão da variável resposta
## t = número de tratamentos
## d = diferença significativa
## ri = número inicial de repetições

df.res<-(ri*t-1)-(ri-1)-(t-1)
q<-qtukey(alpha,t,df.res)
f<-qf(alpha,df.res,n2)
r<-(q^2*s^2*f)/(d^2)
return(r,df.res,q,f)
}
```

Para um experimento conduzido em outro delineamento, baste redefinir *df.res*.

Utilizando a informação do CV

Para calcular o número de repetições através do CV, basta substituir o valor de *s* pelo CV% e o valor de *d* por *d*%.

Exercícios

1. Escolha um conjunto de dados de um experimento das notas de aula e calcule o número de repetições para um novo experimento.
2. Refaça os cálculos utilizando o CV% do mesmo experimento.
3. Utilize diferentes valores de *d*, *s* e *alpha* para visualizar o comportamento do número de repetições.