

Revisão de conhecimento

CE074 - Controle de Processos Industriais

2017/2

1. Quais as suposições de um experimento inteiramente casualizado? As UEs devem ser independentes.

A determinação dos níveis dos fatores para as UEs deve ser aleatória.

A distribuição dos dados amostrais deve ser normal (?).

A variância deve ser homocedástica entre os níveis dos fatores (?).

2. Quais as suposições de um experimento em blocos ao acaso?

Os blocos devem ser homogêneos (as características de dentro dos blocos devem ser semelhantes).

As UEs devem ser independentes.

A determinação dos níveis dos fatores para as UEs dentro dos blocos deve ser aleatória.

A distribuição dos dados amostrais deve ser normal (?).

A variância deve ser homogênea dentro dos blocos (?).

3. O que é um experimento em blocos incompletos?

No DBC o número de UEs por bloco deve ser igual ao número de tratamentos, mas nem sempre isso é possível. Quando não é possível utilizar todos os tratamentos em cada um dos blocos deve-se utilizar um delineamento em blocos incompletos. Por exemplo, podemos ter um fator com 6 níveis, mas devido à alguma restrição (física, logística, orçamentária, ...) temos apenas 4 UEs por bloco.

4. O que é um experimento em blocos incompletos balanceado?

Um bloco incompleto balanceado é um delineamento em que cada um dos níveis do fator é repetido r vezes e ocorre no máximo uma vez em cada um dos b blocos.

Todos os blocos tem o mesmo tamanho, e todos os tratamentos tem o mesmo número de repetições.

5. Por que agrupamos unidades experimentais (UEs) em blocos?

Para fazer o controle local. Agrupar UEs em blocos serve para homogeneizar as amostras em termos de variabilidade, pois o agrupamento é feito para que as amostras dentro de cada bloco possuam características similares.

6. Como se espera que sejam as UEs dentro de um bloco?

Espera-se que as UEs sejam homogêneas, ou seja, que a variabilidade entre as amostras de um mesmo bloco seja pequena. A variabilidade das amostras dentro de um bloco deve ser menor do que a variabilidade entre as amostras se não houvessem os blocos.

7. Por que aleatorizamos os níveis do fator às UE?

Para evitar que ocorram erros sistemáticos nas amostras. Sabemos que mesmo entre UEs aparentemente idênticas, existe uma particularidade natural de cada UE (e.g. plantas, máquinas, ...). A aleatorização entre as UEs garante que os níveis do fator de interesse serão aplicados independentemente destas características, fazendo com que qualquer variação observada entre as UEs seja efeito apenas do nível do fator e da variação aleatória intrínseca ao mecanismo gerador da resposta.

8. Em experimentos agrícolas é comum não se colher a bordadura das parcelas. Em experimentos médicos, médicos e pacientes não sabem que níveis do fator foram administrados. Nas pesquisas de opinião, evita-se que uma pessoa conheça a avaliação das demais. Por que são adotados tais procedimentos?

Para garantir a independência entre as amostras.

9. (Descreva) Como é a aleatorização dos níveis em um experimento inteiramente casualizado?

Os níveis dos fatores são atribuídos aleatoriamente para cada UE. Uma forma simples de fazer este sorteio aleatório é definir um valor para cada UE.

(Use o R) Como você faria se tivesse um fator com 3 níveis (A, B, C), e quisesse fazer 4 repetições para cada nível do fator?

```
## Dessa forma, temos 3 x 4 = 12 UEs. Associamos os valores de 1 a 12
## para cada UE, e sorteamos os níveis dos fatores para cada uma
nf <- 3
r <- 4
n <- nf * r
fator <- rep(c("A", "B", "C"), each = r)
set.seed(123)
(DIC <- data.frame(UE = 1:n, Fator = sample(fator)))
```

```
#   UE Fator
# 1   1     A
# 2   2     C
# 3   3     B
# 4   4     B
# 5   5     C
# 6   6     A
# 7   7     C
# 8   8     C
# 9   9     A
# 10 10     A
# 11 11     B
# 12 12     B
```

```
## Croqui do experimento
matrix(DIC$Fator, ncol = nf, dimnames = list(1:r, 1:nf))
```

```
# 1 2 3
# 1 "A" "C" "A"
# 2 "C" "A" "A"
# 3 "B" "C" "B"
# 4 "B" "C" "B"
```

1. **(Descreva)** Como é a aleatorização dos níveis em um experimento de blocos ao acaso?

Quando temos blocos, a aleatorização é realizada dentro de cada bloco. Portanto precisamos inicialmente definir os blocos e verificar quantas UEs pertencem a cada bloco. Idealmente, cada nível do fator deve estar presente em todos os blocos.

(Use o R) Suponha que temos um experimento com 4 blocos, e um fator com 6 níveis para ser avaliado. Como seria a aleatorização?

```
## Nesse caso, temos 6 x 4 = 24 UEs. Associamos os valores de 1 a 24
## para cada UE, e alocamos cada 6 UEs em cada um dos 4 blocos. Os
## níveis dos fatores devem ser sorteados dentro de cada bloco
nf <- 6
b <- 4
n <- nf * b
fator <- rep(LETTERS[1:6], each = b)
blocos <- rep(c("I", "II", "III", "IV"), each = nf)
set.seed(123)
(DBC <- data.frame(UE = 1:n, Blocos = blocos, Fator = sample(fator)))
```

```
#   UE Blocos Fator
# 1  1      I      B
# 2  2      I      E
# 3  3      I      C
# 4  4      I      F
# 5  5      I      F
# 6  6      I      A
# 7  7     II      C
# 8  8     II      D
# 9  9     II      F
#10 10     II      F
#11 11     II      D
#12 12     II      B
#13 13    III      E
#14 14    III      D
#15 15    III      A
#16 16    III      C
#17 17    III      E
#18 18    III      E
#19 19     IV      B
#20 20     IV      B
#21 21     IV      A
#22 22     IV      A
```

```
# 23 23      IV      D
# 24 24      IV      C
```

Está certo o que foi feito até aqui?

```
set.seed(123)
(DBC <- data.frame(UE = 1:n, Blocos = blocos,
                  Fator = as.vector(
                      replicate(b, sample(unique(fator)))
                  )))
```

```
#      UE Blocos Fator
# 1    1      I      B
# 2    2      I      D
# 3    3      I      F
# 4    4      I      C
# 5    5      I      E
# 6    6      I      A
# 7    7     II     D
# 8    8     II     E
# 9    9     II     C
# 10 10     II     B
# 11 11     II     F
# 12 12     II     A
# 13 13    III     E
# 14 14    III     C
# 15 15    III     A
# 16 16    III     F
# 17 17    III     D
# 18 18    III     B
# 19 19     IV     B
# 20 20     IV     E
# 21 21     IV     D
# 22 22     IV     C
# 23 23     IV     F
# 24 24     IV     A
```

Croqui do experimento

```
matrix(DBC$Fator, ncol = b, dimnames = list(1:nf, paste("Bloco", 1:b)))
```

```
#      Bloco 1 Bloco 2 Bloco 3 Bloco 4
# 1 "B"      "D"      "E"      "B"
# 2 "D"      "E"      "C"      "E"
# 3 "F"      "C"      "A"      "D"
# 4 "C"      "B"      "F"      "C"
# 5 "E"      "F"      "D"      "F"
# 6 "A"      "A"      "B"      "A"
```

1. O que é um experimento fatorial 2^k ?

Experimentos fatoriais são aqueles em que um mais fatores são estudados

simultaneamente. A vantagem é que podemos analisar se existe efeito da interação entre estes fatores, não apenas efeitos individuais. Um fatorial 2^k significa que temos um experimento com $k > 1$ fatores cada um com apenas 2 níveis.

2. **(Descreva)** Estendendo a ideia: como é a aleatorização de um experimento inteiramente ao acaso (DIC) no esquema fatorial 2^k ?

É similar à aleatorização para um DIC com um fator, ou seja sorteamos aleatoriamente as UEs que receberão determinado nível do fator. A grande diferença é que como temos mais de um fator, devemos casualizar os níveis combinados dos fatores.

(Use o R) Suponha que temos um experimento em DIC com 2 níveis do fator A (a), 2 níveis do fator B (b), e sem repetições (r), ou seja, um fatorial 2^2 . Como seria a aleatorização?

```
## Nesse caso temos 2 níveis A x 2 níveis B = 4 possibilidades de
## combinação. Criamos os níveis dos fatores e fazemos todas as
## combinações possíveis. Depois aleatorizamos as combinações para cada
## UE.
a <- 2
b <- 2
r <- 1 # sem repetição
n <- a * b * r
fatores <- expand.grid(A = c("A1", "A2"),
                      B = c("B1", "B2"))
fatores$AB <- paste(fatores$A, fatores$B, sep = ":")
fatores
```

```
#   A B   AB
# 1 A1 B1 A1:B1
# 2 A2 B1 A2:B1
# 3 A1 B2 A1:B2
# 4 A2 B2 A2:B2
```

```
set.seed(123)
(DICf <- data.frame(UE = 1:n, Fatores = sample(fatores$AB)))
```

```
#   UE Fatores
# 1  1  A2:B1
# 2  2  A1:B2
# 3  3  A1:B1
# 4  4  A2:B2
```

```
table(DICf$Fatores)
```

```
#
# A1:B1 A1:B2 A2:B1 A2:B2
#      1      1      1      1
```

```
## Croqui para este experimento
matrix(DICf$Fatores, ncol = a*b,
```

```
dimnames = list(1:r, 1:(a*b)))
```

```
#   1       2       3       4
# 1 "A2:B1" "A1:B2" "A1:B1" "A2:B2"
```

E se houvessem 4 repetições (r) de cada nível do fator?

```
a <- 2
b <- 2
r <- 4
n <- a * b * r
fatores <- expand.grid(A = rep(c("A1", "A2"), each = a),
                      B = rep(c("B1", "B2"), each = b))
fatores$AB <- paste(fatores$A, fatores$B, sep = ":")
fatores
```

```
#   A B  AB
# 1 A1 B1 A1:B1
# 2 A1 B1 A1:B1
# 3 A2 B1 A2:B1
# 4 A2 B1 A2:B1
# 5 A1 B1 A1:B1
# 6 A1 B1 A1:B1
# 7 A2 B1 A2:B1
# 8 A2 B1 A2:B1
# 9 A1 B2 A1:B2
# 10 A1 B2 A1:B2
# 11 A2 B2 A2:B2
# 12 A2 B2 A2:B2
# 13 A1 B2 A1:B2
# 14 A1 B2 A1:B2
# 15 A2 B2 A2:B2
# 16 A2 B2 A2:B2
```

```
set.seed(123)
(DICf <- data.frame(UE = 1:n, Fatores = sample(fatores$AB)))
```

```
#   UE Fatores
# 1   1  A1:B1
# 2   2  A2:B2
# 3   3  A1:B1
# 4   4  A2:B2
# 5   5  A1:B2
# 6   6  A1:B1
# 7   7  A1:B2
# 8   8  A1:B2
# 9   9  A2:B2
# 10 10  A2:B1
# 11 11  A1:B2
# 12 12  A2:B1
```

```
# 13 13    A2:B1
# 14 14    A1:B1
# 15 15    A2:B2
# 16 16    A2:B1
```

```
table(DICf$Fatores)
```

```
#
# A1:B1 A1:B2 A2:B1 A2:B2
#      4      4      4      4
```

```
## Croqui para este experimento
matrix(DICf$Fatores, ncol = a*b,
       dimnames = list(1:r, 1:(a*b)))
```

```
#   1      2      3      4
# 1 "A1:B1" "A1:B2" "A2:B2" "A2:B1"
# 2 "A2:B2" "A1:B1" "A2:B1" "A1:B1"
# 3 "A1:B1" "A1:B2" "A1:B2" "A2:B2"
# 4 "A2:B2" "A1:B2" "A2:B1" "A2:B1"
```

1. Einstein deseja avaliar a qualidade do café processado de duas maneiras diferentes (A e B). No sábado ele recebeu a visita de 4 amigos e serviu o café A para eles registrando as avaliações feitas. No domingo, sua esposa recebeu a visita de 5 amigas e serviu o café B para elas registrando as avaliações. Existem problemas com o experimento de Einstein? Identifique-os. Aconselhe-o de como realizar o experimento para investigar sua hipótese.

O problema neste experimento é que os efeitos estão confundidos. O tipo de café está confundido com o dia, com o sexo das pessoas, e com a pessoa que preparou o café. Uma análise dos dados deste experimento não será muito informativa pois não conseguiremos separar os efeitos de todos os fatores.

Uma análise para avaliar a qualidade do café poderia ser feita da seguinte forma: Apenas Einstein prepara o café (fixa, ou controla, o preparador). Ele prepara os cafés A e B para um grupo de homens e mulheres em um dia. No outro dia ele prepara novamente os tipos A e B para outro grupo de homens e mulheres. Nesse caso, os dias são os blocos, pois o interesse da análise é na qualidade de cada tipo de café avaliada por homens e mulheres.

2. O que são variáveis *dummy*?

São variáveis que servem para codificar a presença (1) ou ausência (0) do nível de algum fator em uma determinada UE.

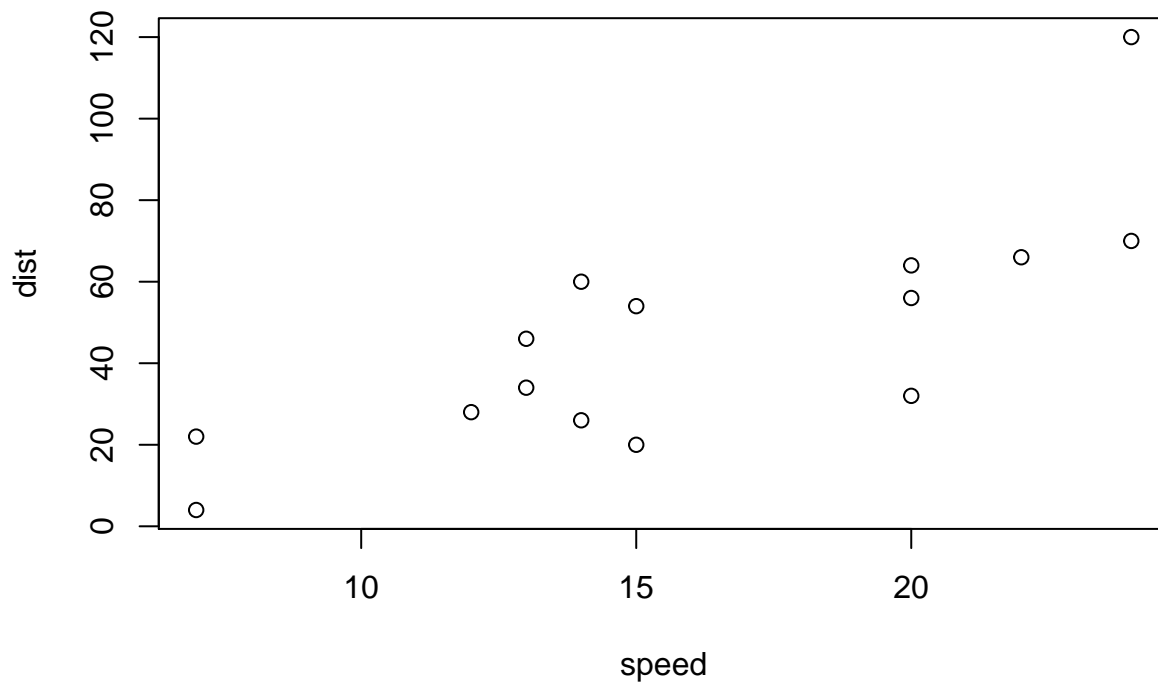
3. Qual a diferença na matriz do modelo entre um modelo de regressão (fatores de níveis métricos) e modelo de experimento (fatores de níveis categóricos)?

A matriz do modelo de um modelo de regressão contém os próprios valores mensurados da variável resposta, enquanto que a matriz do modelo de um experimento com um fator contém variáveis *dummy* para indicar a presença/ausência dos níveis do fator.

```
## Um exemplo com cars
set.seed(123)
(carros <- cars[sample(1:nrow(cars), size = 15), ])
```

```
#   speed dist
# 15    12   28
# 39    20   32
# 20    14   26
# 42    20   56
# 44    22   66
# 3     7    4
# 24    15   20
# 49    24  120
# 46    24   70
# 19    13   46
# 43    20   64
# 18    13   34
# 26    15   54
# 22    14   60
# 4     7    22
```

```
plot(dist ~ speed, data = carros)
```



```
model.matrix(dist ~ speed, data = carros)
```

```
#   (Intercept) speed
# 15           1    12
# 39           1    20
# 20           1    14
```



```
# 42      1    20
# 44      1    22
# 3       1     7
# 24      1    15
# 49      1    24
# 46      1    24
# 19      1    13
# 43      1    20
# 18      1    13
# 26      1    15
# 22      1    14
# 4       1     7
# attr(,"assign")
# [1] 0 1
```

```
## Um fator com 3 níveis e 4 repetições
fator <- rep(c("A", "B", "C"), each = 4)
model.matrix(~fator)
```

```
#      (Intercept) fatorB fatorC
# 1             1      0      0
# 2             1      0      0
# 3             1      0      0
# 4             1      0      0
# 5             1      1      0
# 6             1      1      0
# 7             1      1      0
# 8             1      1      0
# 9             1      0      1
# 10            1      0      1
# 11            1      0      1
# 12            1      0      1
# attr(,"assign")
# [1] 0 1 1
# attr(,"contrasts")
# attr(,"contrasts")$fator
# [1] "contr.treatment"
```

1. Por que precisamos adotar uma restrição paramétrica nos modelos com fator de nível categórico?

Porque a matriz do modelo com variáveis dummy de todos os níveis do fator possui posto incompleto, ou seja, não é inversível. Como sabemos que precisamos da inversa de $(X'X)$, então é necessário impor uma restrição para que essa matriz possa ser invertida, e assim podemos estimar os parâmetros do modelo.

```
## Fator com 3 níveis e 4 repetições
fator <- factor(rep(c("A", "B", "C"), each = 4))
## Cria a matriz do modelo sem nenhuma restrição
```

```
X <- matrix(0, nrow = 12, ncol = 3)
X[cbind(seq_along(fator), fator)] <- 1
(X <- cbind(1, X))
```

```
#      [,1] [,2] [,3] [,4]
# [1,]    1    1    0    0
# [2,]    1    1    0    0
# [3,]    1    1    0    0
# [4,]    1    1    0    0
# [5,]    1    0    1    0
# [6,]    1    0    1    0
# [7,]    1    0    1    0
# [8,]    1    0    1    0
# [9,]    1    0    0    1
# [10,]   1    0    0    1
# [11,]   1    0    0    1
# [12,]   1    0    0    1
```

```
## X'
Xt <- t(X)
## X'X
(Xt %*% X)
```

```
#      [,1] [,2] [,3] [,4]
# [1,]   12    4    4    4
# [2,]    4    4    0    0
# [3,]    4    0    4    0
# [4,]    4    0    0    4
```

```
## (X'X)^-1
solve(Xt %*% X)
```

```
# Error in solve.default(Xt %*% X): system is computationally singular: reciprocal condition
```

```
## Impondo uma restrição: remove a coluna do primeiro nível -
## equivalente a zerar o primeiro nível do fator
(X <- X[, -2])
```

```
#      [,1] [,2] [,3]
# [1,]    1    0    0
# [2,]    1    0    0
# [3,]    1    0    0
# [4,]    1    0    0
# [5,]    1    1    0
# [6,]    1    1    0
# [7,]    1    1    0
# [8,]    1    1    0
# [9,]    1    0    1
# [10,]   1    0    1
# [11,]   1    0    1
```

```
# [12,]    1    0    1
```

```
## X'
```

```
Xt <- t(X)
```

```
## X'X
```

```
Xt %*% X
```

```
#      [,1] [,2] [,3]
```

```
# [1,]   12    4    4
```

```
# [2,]    4    4    0
```

```
# [3,]    4    0    4
```

```
## (X'X)^-1
```

```
solve(Xt %*% X)
```

```
#      [,1] [,2] [,3]
```

```
# [1,]  0.25 -0.25 -0.25
```

```
# [2,] -0.25  0.50  0.25
```

```
# [3,] -0.25  0.25  0.50
```

```
MASS::fractions(solve(Xt %*% X))
```

```
#      [,1] [,2] [,3]
```

```
# [1,]  1/4 -1/4 -1/4
```

```
# [2,] -1/4  1/2  1/4
```

```
# [3,] -1/4  1/4  1/2
```

1. Por que fazemos análise dos resíduos?

Para verificar as suposições feitas para o modelo.