

Curso R

Ecologia

Alexandre Adalardo de Oliveira

IBUSP maio 2017

Uso da Linguagem R

Aula 3

Classes de Objetos

- Vetores
- Dataframe
- Matrix
- Lista

Importando dados

Um problema



Importando datos

trapa.xls (Modo de Compatibilidade) - Microsoft Excel

Inserir Layout da Página Fórmulas Dados Revisão

De Texto De Outras Fontes Conexões Existentes Atualizar tudo Conexões

Conexões Propriedades Editar Links Class

A1 fx

B	C	D	E	F	G
nascimento	estado	vivo	altura		
1936	CE	s	1.68		
1936	RJ	s	1.79		
1941	RJ	n	1.81		
1934	MG	n	1.61		

trapa.xls

Ale_2009 > AulasUSP > Introd_R > Aulas > dados

Organize Views New Folder

Favorite Links

Documents

More >>

Folders

- AulasUSP
 - bie5786
 - CursoCampo
 - EcoPop
 - Introd_R
 - Aulas
 - dados
 - intrn_R.nnt

Nome do arquivo: trapa.xls

Tipo: Pasta de Trabalho do Excel 97-2003 (*.xls)

Autores:

- Pasta de Trabalho do Excel (*.xlsx)
- Pasta de Trabalho Habilitada para Macro do Excel (*.xlsm)
- Pasta de Trabalho Binária do Excel (*.xlsb)
- Pasta de Trabalho do Excel 97-2003 (*.xls)
- Dados XML (*.xml)
- Página da Web de Arquivo Único (*.mht;*.mhtml)
- Página da Web (*.htm;*.html)
- Modelo do Excel (*.xlt)
- Modelo Habilitado para Macro do Excel (*.xltn)
- Modelo do Excel 97-2003 (*.xlt)
- Texto (separado por tabulações) (*.txt)
- Texto em Unicode (*.txt)
- Planilha XML 2003 (*.xml)
- Pasta de trabalho do Microsoft Excel 5.0/95 (*.xls)
- CSV (separado por vírgulas) (*.csv)
- Texto formatado (separado por espaços) (*.prn)
- Texto (Macintosh) (*.txt)
- Texto (MS-DOS) (*.txt)
- CSV (Macintosh) (*.csv)
- CSV (MS-DOS) (*.csv)
- DIF (Formato de troca de dados) (*.dif)
- SYLK (vínculo simbólico) (*.slk)
- Suplemento do Excel (*.xlam)
- Suplemento do Excel 97-2003 (*.xla)

Ocultar pastas

Inserir Layout da Página Fórmulas **Dados** Revisão Exibição Suplementos

De Texto De Outras Fontes Conexões Existentes Atualizar tudo Conexões Classificar Filtro Limpar Reaplicar Avançado Texto para colunas Remover Duplicatas Validação de Dados Consolidar Teste de Hipóteses Agrupar Desagrupar Subtotal Estrutura de Tópicos Análise Solução

Classificar e Filtrar Ferramentas de Dados Estrutura de Tópicos

A1

B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
nascimento	estado	vivo	altura													
1936	CE	s	1.68													
1936	RJ	s	1.79													
1941	RJ	n	1.81													
1934	MG	n	1.61													

trapa.txt - Notepad

File Edit Format View Help

```

nascimento      estado      vivo      altura
Didi 1936 CE s 1.68
Dedé 1936 RJ s 1.79
Mussum 1941 RJ n 1.81
Zacarias 1934 MG n 1.61

```

trapa.csv - Notepad

File Edit Format View Help

```

,nascimento,estado,vivo,altura
Didi,1936,CE,s,1.68
Dedé,,1936,RJ,s,1.79
Mussum,1941,RJ,n,1.81
Zacarias,1934,MG,n,1.61

```

Importando arquivo texto

Diretório de trabalho

```
> getwd()
```

```
## [1] "/home/aao/Ale2016/AleCursos/RUSE"
```

```
> dir()
```

```
## [1] "aula3.html" "aula3.r"  
## [3] "aula3.rmd~" "dados"  
## [5] "scriptRAulaDado2016.R" "script"
```

```
> dir("dados")
```

```
## [1] "gua.clas.txt" "trapa_nome.t  
## [4] "trapa_nome1.txt" "trapa.csv"
```

```
## [7] "trapa.xls" "trapa1.csv"  
## [10] "trapaVirgula.txt"
```

Função de leitura de dados

```
> read.table("dados/trapa.csv", header=T
```

```
##      codinome  nascimento  estado  vivo  alt
## 1      Didi      1936      CE      s     1
## 2      Dede      1936      RJ      s     1
## 3      Mussum    1941      RJ      n     1
## 4 Zacarias    1934      MG      n     1
```

```
> ls()
```

```
## character(0)
```

Atribuição a um objeto

```
> trapa <- read.table("dados/trapa.csv",  
> ls())
```

```
## [1] "trapa"
```

```
> trapa
```

```
##      codinome  nascimento  estado  vivo  alt  
## 1      Didi      1936      CE      s     1  
## 2      Dede      1936      RJ      s     1  
## 3      Mussum    1941      RJ      n     1  
## 4 Zacarias    1934      MG      n     1
```


LEARNING FOR A COMPLEX WORLD



Mundo Complexo

- Dependência do idioma (locale)
 - decimal (“.” ou “,”)
 - separador de campo (“,”, “;”, “tab”)
- Codificação de fontes
 - caracteres especiais (acentos)
- `read.table` é flexível
- leitura limpa não pressupõem correção

Leitura: problemas

```
> (trapa1 <- read.table("dados/trapa.csv"))

##      codinome.nascimento.estado.vivo.alt
## 1          Didi,1936,CE,s,1
## 2          Dede,1936,RJ,s,1
## 3      Mussum,1941,RJ,n,1
## 4      Zacarias,1934,MG,n,1

> str(trapa1)

## 'data.frame':    4 obs. of  1 variable:
##  $ codinome.nascimento.estado.vivo.al
```

Leitura: problemas

```
> (trapa2 <- read.table("dados/trapaVirg

##                               codinome.nascime
## Didi;1936;CE;s;1
## Dede;1936;RJ;s;1
## Mussum;1941;RJ;n;1
## Zacarias;1934;MG;n;1

> str(trapa2)

## 'data.frame':    4 obs. of  1 variabl
## $ codinome.nascimento.estado.vivo.al
```

```
> (trapa2 <- read.table("dados/trapaVirg
```

```
##      codinome  nascimento  estado  vivo  alt
## 1      Didi      1936        CE      s    1
## 2      Dede      1936        RJ      s    1
## 3      Mussum    1941        RJ      n    1
## 4 Zacarias    1934        MG      n    1
```

```
> str(trapa2)
```

```
## 'data.frame':    4 obs. of  5 variabl
## $ codinome      : Factor w/ 4 levels "De
## $ nascimento: int  1936 1936 1941 19
## $ estado       : Factor w/ 3 levels "CE
## $ vivo        : Factor w/ 2 levels "n"
## $ altura      : Factor w/ 4 levels "1,
```

```
> (trapa2 <- read.table("dados/trapaVirg
```

```
##      codinome  nascimento  estado  vivo  alt
## 1      Didi      1936      CE      s     1
## 2      Dede      1936      RJ      s     1
## 3      Mussum    1941      RJ      n     1
## 4 Zacarias    1934      MG      n     1
```

```
> str(trapa2)
```

```
## 'data.frame':    4 obs. of  5 variabl
## $ codinome      : Factor w/ 4 levels "De
## $ nascimento: int  1936 1936 1941 19
## $ estado       : Factor w/ 3 levels "CE
## $ vivo        : Factor w/ 2 levels "n"
## $ altura      : num  1.68 1.79 1.81 1.
```

```
> (trapa2 <- read.table("dados/trapaVirg
```

```
##      codinome  nascimento  estado  vivo  alt
## 1      Didi      1936      CE      s      1
## 2      Dede      1936      RJ      s      1
## 3      Mussum    1941      RJ      n      1
## 4 Zacarias    1934      MG      n      1
```

```
> str(trapa2)
```

```
## 'data.frame':    4 obs. of  5 variabl
## $ codinome      : chr  "Didi" "Dede" "Mu
## $ nascimento    : int   1936 1936 1941 19
## $ estado        : chr  "CE" "RJ" "RJ" "M
## $ vivo          : chr  "s" "s" "n" "n"
## $ altura        : num  1.68 1.79 1.81 1.
```

Help!

```
> ?read.table
```

Evitando transtornos

- separação de campo
- evite usar “,” como decimal
 - mude a configuração do computador
- estude os argumentos do `read.table`
- cuidado com caracteres especiais (’, “, ’, `)
- use encoding UTF8

Estabeleça um padrão

```
> trapalhao <- read.table("dados/trapa_n
```

Padrão Alê

- nome de variáveis simples
- metadado para explicar a variável
- separador de campo tabulação
- decimal “.” PLEASE!
- leitura de caracteres (as.is)

Dados de outras fontes

- abra no libreOffice
- salve no seu padrão de texto
- use encoding UTF8
- substitua ‘`’ " se necessário

Pós leitura

```
> str(trapalhao)
```

```
## 'data.frame':      4 obs. of  6 variables
## $ codinome      : chr  "Didi" "Dedé" "Mu
## $ nascimento    : int   1936  1936  1941  19
## $ estado        : chr   "CE"  "RJ"  "RJ"  "M
## $ vivo          : chr   "s"   "s"   "n"   "n"
## $ altura        : num   1.68  1.79  1.81  1.
## $ nome          : chr   "Antônio Renato A
```

```
> dim(trapa)
```

```
## [1] 4 5
```

Pós leitura

FAÇA UMA ANÁLISE EXPLORATÓRIA DOS DADOS

AULA DE SEXTA-FEIRA!



Manipulando objetos



Vetores

```
> (v1 <- seq(from = 0, to = 1000, by = 1
```

```
## [1] 0 100 200 300 400 500 6
```

```
> class(v1)
```

```
## [1] "numeric"
```

```
> length(v1)
```

```
## [1] 11
```

```
> str(v1)
```

```
## num [1:11] 0 100 200 300 400 500 600
```

```
> v1/100
```

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


Vetores

```
> str(v1)
```

```
##   num [1:11] 0 100 200 300 400 500 600
```

```
> v1/100
```

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

Extração

```
> v1[5]
```

```
## [1] 400
```

```
> v1[2:7]
```

```
## [1] 100 200 300 400 500 600
```

```
> v1[c(1, 3, 5, 7, 11)]
```

```
## [1] 0 200 400 600 1000
```

```
> v1[seq(1, 11, by=2)]
```

```
## [1] 0 200 400 600 800 1000
```

```
> seq(1, 11, by=2)
```

```
## [1] 1 3 5 7 9 11
```



```
> v1/100
#Error in v1/100 : non-numeric argument
> class(v1)
#[1] "character"
```

```
> v1[5] <- 400
```

```
> v1[5]
```

```
## [1] "400"
```

```
> v1
```

```
## [1] "0" "100" "200" "300" "400"
```

```
## [11] "1000"
```

```
> class(v1)
```

```
## [1] "character"
```

Funções de coerção

```
> is.numeric(v1)

## [1] FALSE

> v1 <- as.numeric(v1)
> is.numeric(v1)

## [1] TRUE
```



Classes de objetos

Linguagem orientada a objetos

```
> area <- c(303, 379, 961, 295, 332, 47,
> area

## [1] 303 379 961 295 332 47 1

> class(area)

## [1] "numeric"

> summary(area)

##      Min. 1st Qu.  Median    Mean 3rd Q
## 11.00  70.25  299.00  525.20  367.
```

```

> riqueza <- c(3, 10, 20, 7, 8, 4, 8, 3)
> modelo1 <- lm(riqueza~area)
> class(modelo1)

## [1] "lm"

> summary(modelo1)

##
## Call:
## lm(formula = riqueza ~ area)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.4614 -2.1245 -0.5101  1.4862  7.68
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)

```

Dataframe

Qual a classe do objeto trapa?

```
> class(trapa)
```

```
## [1] "data.frame"
```

```
> str(trapa)
```

```
## 'data.frame': 4 obs. of 5 variables:
## $ codinome : chr "Didi" "Dede" "Mu
## $ nascimento: int 1936 1936 1941 19
## $ estado : chr "CE" "RJ" "RJ" "M
## $ vivo : chr "s" "s" "n" "n"
## $ altura : num 1.68 1.79 1.81 1.
```

```
> names (trapa)
## [1] "codinome" "nascimento" "estadc

> colnames (trapa)
## [1] "codinome" "nascimento" "estadc

> names (trapa) [5]
## [1] "altura"

> names (trapa) [5] <- "alt"
> names (trapa) [5]
## [1] "alt"
```

Indexadores de data.frame

- \$: indexa o nome da coluna
- colchete : linha, coluna

```
> trapa$codinome
```

```
## [1] "Didi"      "Dede"      "Mussum"
```

```
> trapa$codinome[2]
```

```
## [1] "Dede"
```

```
> trapa$codinome[c(1,2)]
```

```
## [1] "Didi" "Dede"
```

Duas dimensões

```
> trapa$codinome[2]
```

```
## [1] "Dede"
```

```
> trapa$codinome[c(2,1)]
```

```
## [1] "Dede" "Didi"
```

```
> trapa[2,1]
```

```
## [1] "Dede"
```

Coerção e sobrescrição

```
> trapa$vivo
```

```
## [1] "s" "s" "n" "n"
```

```
> trapa$vivo <- c(1,1,0,0)
```

```
> str(trapa$vivo)
```

```
##   num [1:4] 1 1 0 0
```

```
> trapa$vivo <- as.logical(trapa$vivo)
```

```
> trapa$vivo
```

```
## [1] TRUE TRUE FALSE FALSE
```


Novas variáveis

```
> trapa$nascimento
```

```
## [1] 1936 1936 1941 1934
```

```
> 2017 - trapa$nascimento
```

```
## [1] 81 81 76 83
```

```
> trapa$idade <- 2017 - trapa$nascimento
```

```
> trapa$idade
```

```
## [1] 81 81 76 83
```

Dataframe

Classe dataframe

- vetores de mesmo tamanho
- concatenados
- natureza de variáveis diferentes
- UMA PLANILHA DADOS

Matrix

```
> matrix(1:12, nrow=4, ncol=3)
```

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

Matrix

```
> matrix(1:12, nrow=4, ncol=3, byrow=TRUE)
```

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

Matrix

```
> (ilhas = matrix(sample(c(round(runif(2
```

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

Como assim?

```
> round(c(runif(20, 0, 6), rep(0, 20)))
```

```
## [1] 6 2 4 2 1 1 4 4 5 5 0 2 5 2 3 4  
## [36] 0 0 0 0 0
```

```
> sample(round(c(runif(36, 0, 6), rep(0, 4)))
```

```
## [1] 0 4 1 0 2 4 3 0 3 4 3 6 3 5 5 2  
## [36] 0 5 6 4 4
```

Nomeando Linhas

```
> paste("ilha", 1:8, sep="_")  
## [1] "ilha_1" "ilha_2" "ilha_3" "ilha_  
> rownames(ilhas) <- paste("ilha", 1:8, s
```

Nomeando colunas

```
> colnames(ilhas) <- paste("sp", 1:5, sep="")  
> ilhas
```

```
##           sp_1 sp_2 sp_3 sp_4 sp_5  
## ilha_1      0   3   0   0   0  
## ilha_2      0   4   4   3   2  
## ilha_3      0   0   2   5   0  
## ilha_4      5   3   0   0   3  
## ilha_5      0   6   4   0   3  
## ilha_6      3   0   0   0   1  
## ilha_7      0   6   5   2   3  
## ilha_8      0   6   0   0   0
```


Indexando

```
> ilhas[,1]
```

```
## ilha_1 ilha_2 ilha_3 ilha_4 ilha_5 il  
##      0      0      0      5      0
```

```
> ilhas[, "sp_5"]
```

```
## ilha_1 ilha_2 ilha_3 ilha_4 ilha_5 il  
##      0      2      0      3      3
```

```
> ilhas[, c(2, 4, 5)]
```

```
##          sp_2 sp_4 sp_5  
## ilha_1      3  0  0  
## ilha_2      4  3  2  
## ilha_3      0  5  0
```

```
## ilha_4      3      0      3
## ilha_5      6      0      3
## ilha_6      0      0      1
## ilha_7      6      2      3
## ilha_8      6      0      0
```

```
> ilhas[, c("sp_2", "sp_4", "sp_5")]
```

```
##           sp_2 sp_4 sp_5
## ilha_1      3   0   0
## ilha_2      4   3   2
## ilha_3      0   5   0
## ilha_4      3   0   3
## ilha_5      6   0   3
## ilha_6      0   0   1
## ilha_7      6   2   3
## ilha_8      6   0   0
```

Matrizes

- linhas e colunas concatenados
- vetores de mesma natureza
- operam algebra linear

Lista

```
> (a=1:5)
```

```
## [1] 1 2 3 4 5
```

```
> (b=factor(rep(c("a", "b", "c"), each=3)) )
```

```
## [1] a a a b b b c c c
```

```
## Levels: a b c
```

```
> (c=data.frame(sec=c("XIX", "XX", "XXI")
```

```
##      sec inicio
```

```
## 1 XIX      1801
```

```
## 2  XX      1901
```

```
## 3 XXI      2001
```

```
> m1 = list(v1 = a, f1 = b, df1 = c)
```

```
> m1
```

```
## $v1
```

```
## [1] 1 2 3 4 5
```

```
##
```

```
## $f1
```

```
## [1] a a a b b b c c c
```

```
## Levels: a b c
```

```
##
```

```
## $df1
```

```
##      sec inicio
```

```
## 1  vtv      1001
```

Indexando Listas

```
> ml$df1$inicio
```

```
## [1] 1801 1901 2001
```

```
> ml$df1$inicio <- c(1800, 1900, 2000)
```

```
> ml$df1$inicio
```

```
## [1] 1800 1900 2000
```

Modificando Listas

```
> ml$inicio<-"início do século"  
> str(ml)  
  
## List of 4  
## $ v1      : int [1:5] 1 2 3 4 5  
## $ f1      : Factor w/ 3 levels "a","b"  
## $ df1     : 'data.frame':  3 obs. of  2  
## ..$ sec   : Factor w/ 3 levels "XIX  
## ..$ inicio: num [1:3] 1800 1900 200  
## $ inicio: chr "início do século"
```

Indexação de listas

```
> m1$f1
```

```
## [1] a a a b b b c c c  
## Levels: a b c
```

```
> m1[[2]]
```

```
## [1] a a a b b b c c c  
## Levels: a b c
```

```
> m1[["f1"]]
```

```
## [1] a a a b b b c c c  
## Levels: a b c
```


Indexação de listas

```
> ml$f1[4]
```

```
## [1] b
```

```
## Levels: a b c
```

```
> ml[[3]][1,2]
```

```
## [1] 1800
```

```
> ml[[3]]
```

```
##      sec inicio
```

```
## 1 XIX    1800
```

```
## 2  XX    1900
```

```
## 3 XXI   2000
```

Principais Classes

- vetores [1]
- data.frame [l,c]
- listas [...]
- arrays [l, c, p,...]
 - matrizes [l, c]

Indexadores

```
> alt = c(1.85, 1.78, 1.92, 1.63, 1.81,  
> alt
```

```
## [1] 1.85 1.78 1.92 1.63 1.81 1.55
```

```
> sex=factor(rep(c("M", "F"), each=3))  
> sex
```

```
## [1] M M M F F F  
## Levels: F M
```

Ordenadores de posição

```
> alt
```

```
## [1] 1.85 1.78 1.92 1.63 1.81 1.55
```

```
> rank(alt)
```

```
## [1] 5 3 6 2 4 1
```

```
> sort(alt)
```

```
## [1] 1.55 1.63 1.78 1.81 1.85 1.92
```

```
> order(alt)
```

```
## [1] 6 4 2 5 1 3
```

Indexadores de posição

```
> alt
```

```
## [1] 1.85 1.78 1.92 1.63 1.81 1.55
```

```
> sort(alt)
```

```
## [1] 1.55 1.63 1.78 1.81 1.85 1.92
```

```
> alt[order(alt)]
```

```
## [1] 1.55 1.63 1.78 1.81 1.85 1.92
```

Indexadores de posição

```
> alt
```

```
## [1] 1.85 1.78 1.92 1.63 1.81 1.55
```

```
> sex
```

```
## [1] M M M F F F
```

```
## Levels: F M
```

```
> sex[order(alt)]
```

```
## [1] F F M F M M
```

```
## Levels: F M
```

Operadores lógicos

">", "<", ">=", "<=", "==", "!=", "&"

Indexação com Lógica

```
> haltos <- alt >= 1.80 & sex == "M"
```

```
> haltos
```

```
## [1] TRUE FALSE TRUE FALSE FALSE FAI
```

```
> alt[haltos]
```

```
## [1] 1.85 1.92
```

Quais são homens altos?

```
> which(haltos)
```

```
## [1] 1 3
```

```
> sex
```

```
## [1] M M M F F F
```

```
## Levels: F M
```

```
> alt
```

```
## [1] 1.85 1.78 1.92 1.63 1.81 1.55
```


Familia apply

```
> alt
```

```
## [1] 1.85 1.78 1.92 1.63 1.81 1.55
```

```
> sex
```

```
## [1] M M M F F F
```

```
## Levels: F M
```

```
> tapply(alt, sex, mean)
```

```
##           F           M
```

```
## 1.663333 1.850000
```

Dados Ilha

```
> ilhas
```

```
##           sp_1 sp_2 sp_3 sp_4 sp_5
## ilha_1      0   3   0   0   0
## ilha_2      0   4   4   3   2
## ilha_3      0   0   2   5   0
## ilha_4      5   3   0   0   3
## ilha_5      0   6   4   0   3
## ilha_6      3   0   0   0   1
## ilha_7      0   6   5   2   3
## ilha_8      0   6   0   0   0
```

Quantos indivíduos em cada ilha?

```
> apply(ilhas, 2, sum)
```

```
## sp_1 sp_2 sp_3 sp_4 sp_5  
##      8  28  15  10  12
```

Quantas espécies em cada ilha?

```
> (ilhasvf <- ilhas>0)
```

```
##           sp_1  sp_2  sp_3  sp_4  sp_5
## ilha_1 FALSE  TRUE  FALSE FALSE FALSE
## ilha_2 FALSE  TRUE  TRUE  TRUE  TRUE
## ilha_3 FALSE FALSE  TRUE  TRUE  FALSE
## ilha_4  TRUE  TRUE  FALSE FALSE  TRUE
## ilha_5 FALSE  TRUE  TRUE  FALSE  TRUE
## ilha_6  TRUE  FALSE FALSE  FALSE  TRUE
## ilha_7 FALSE  TRUE  TRUE  TRUE  TRUE
## ilha_8 FALSE  TRUE  FALSE FALSE  FALSE
```

Quantas espécies em cada ilha?

```
> apply(ilhasvf, 1, sum)
```

```
## ilha_1 ilha_2 ilha_3 ilha_4 ilha_5 il  
##      1      4      2      3      3
```

Quanto ocorrências para cada sp

```
> apply(ilhasvf, 2, sum)
```

```
## sp_1 sp_2 sp_3 sp_4 sp_5  
##      2      6      4      3      5
```



Um problema

Guapira

```
> gua.clas=read.table("dados/gua.clas.tx")
> str(gua.clas)
```

```
## 'data.frame':      2191 obs. of  9 vari
## $ floresta      : chr  "tardia" "ta
## $ especie       : chr  "G" "G" "G"
## $ placa         : int   4  7  8 10 11
## $ medida        : chr  "dap" "dap"
## $ tamanho2007   : num   60 71.6 19.2
## $ tamanho2008   : num   63.2 72.9 19
## $ Reprodutivo2007: int    0  0  0  0  0  0
## $ classe2007    : int    3  3  3  3  3  3
## $ classe2008    : int    3  3  3  3  3  3
```

```
> (nivel=unique(gua.clas[,"floresta"]))  
## [1] "tardia" "inicial"  
  
> table(gua.clas$floresta)  
  
##  
## inicial    tardia  
##      1279      912
```

```
> (n0=table(gua.clas[, "classe2007"], gua.
```

```
##
```

```
##      inicial  tardia
```

```
##    1      136     95
```

```
##    2      399    449
```

```
##    3      744    368
```

```
> (trans.abs=table(gua.clas$classe2008 ,  
## , , = inicial  
##  
##  
##      1      2      3  
##  1 105      0      0  
##  2  20 380      0  
##  3   0  11 740  
##  
## , , = tardia  
##
```

```
> (inicial=t(t(trans.abs[, , 1])/n0[, 1]))
```

```
##
```

```
##           1           2           3
```

```
## 1 0.77205882 0.00000000 0.00000000
```

```
## 2 0.14705882 0.95238095 0.00000000
```

```
## 3 0.00000000 0.02756892 0.99462366
```

```
> (tardia=t(t(trans.abs[, , 2])/n0[, 2]))
```

```
##
```

```
##           1           2           3
```

```
## 1 0.7578947 0.00000000 0.00000000
```

```
## 2 0.2105263 0.9599109 0.00000000
```

```
## 3 0.00000000 0.0311804 1.00000000
```