



Instruções:

- Você pode consultar a internet e o cartão de referência do R. Você não pode consultar o caderno nem trocar informações com colegas;
- Todas as questões devem ser respondidas na folha de prova, com os códigos empregados e os resultados obtidos;
- É altamente recomendável que você faça a prova no computador do laboratório (linux). Isso porque a importação de dados pode gerar problemas devido a codificação de caracteres. Caso faça a prova no seu computador pessoal, a responsabilidade de fazer a importação total e correta dos conjuntos de dados é sua.

1. Carregue a área de trabalho e descreva os objetos x, y, z e w em termos de classe, dimensões e tipo de dado armazenado. Passe para o papel o código usado.

```
source("http://www.leg.ufpr.br/~walmes/ensino/ce223-2011-01/ex1.R")

> #-----
> # carregando a area de trabalho
> source("http://www.leg.ufpr.br/~walmes/ensino/ce223-2011-01/ex1.R")
> ls() # lista os objetos criados ao carregar a fonte

[1] "w" "x" "y" "z"

> #-----
> # str() mostra a classe de objeto, dimensão, tipo de dado armazenado
> str(x) # é vetor com 133 elementos do tipo numérico
  num [1:133] 0.745 0.553 0.787 0.515 0.784 ...
> str(y) # é matriz de dimensão 12x12 e armazena caracteres, nomes de cores
  chr [1:12, 1:12] "deepskyblue1" "mediumorchid" "ivory" ...
> str(z) # é um data.frame 150x5, 4 colunas numéricas e 1 fator de 3 níveis
'data.frame':
  150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species      : Factor w/ 3 levels "setosa","versicolor",...: 1 1 1 1 1 1 1 1 1 1 ...
> str(w) # é uma lista de 3 slots, em cada slot tem um dos objetos acima citados

List of 3
 $ x: num [1:133] 0.745 0.553 0.787 0.515 0.784 ...
 $ y: chr [1:12, 1:12] "deepskyblue1" "mediumorchid" "ivory" "darkslategray3" ...
 $ z:'data.frame':
  150 obs. of  5 variables:
  ..$ Sepal.Length: num [1:150] 5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
  ..$ Sepal.Width : num [1:150] 3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
  ..$ Petal.Length: num [1:150] 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
  ..$ Petal.Width : num [1:150] 0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
  ..$ Species      : Factor w/ 3 levels "setosa","versicolor",...: 1 1 1 1 1 1 1 1 1 1 ...
> #-----
```

2. Dos objetos explorados em no exercício 1, obtenha o que se pede. Passe para o papel o código usado.

- a) qual o elemento da 6ª linha e 3ª coluna de y?
- b) qual a média dos valores de x nas posições 5, 33, 24, 67, 109 e 51?
- c) qual a variância dos valores da 3ª coluna de z sem os 22 primeiros elementos?
- d) substitua o primeiro slot de w pelo conteúdo do objeto x com valores ordenados.
- e) quais o 3 últimos elementos do primeiro slot do objeto w?

```
> #-----
> # elemento da 6ª linha e 3ª coluna de y
> y[6,3]

[1] "darkolivegreen2"

> #-----
> # média dos valores de x nas posições 5, 33, 24, 67, 109 e 51
> mean(x[c(5,33,24,67,109,51)])

[1] 0.7008166
```

```

> #-----
> # variância dos valores da 3ª coluna de z sem os 22 primeiros elementos
> var(z[-(1:22), 3])
[1] 2.571001
> #-----
> # substitua o primeiro slot de w pelo conteúdo do objeto x com valores ordenados
> w[[1]] <- sort(x)
> #-----
> # quais o 3 últimos elementos do primeiro slot do objeto w
> length(w[[1]])
[1] 133
> w[[1]][131:133]
[1] 0.9286680 0.9453277 0.9615913
> # ou ainda de uma mais direta
> n <- length(w[[1]])
> w[[1]][(n-2):n]
[1] 0.9286680 0.9453277 0.9615913
> #-----

```

3. Com o objeto `colors()` obtenha o que se pede. Passe para o papel o código usado.

- qual a dimensão do objeto?
- qual a dimensão do objeto que contém apenas os elementos nas posições múltiplas de 7?
- qual a dimensão do objeto que contém apenas os elementos nas posições múltiplas de 7 e 11?
- qual a dimensão do objeto que contém apenas os elementos nas posições múltiplas de 7 ou 11?
- quantos elementos possuem a palavra “red” no nome?
- quantos elementos não possuem a palavra “blue” no nome?

```

> #-----
> # passar colors() para um objeto com nome curto
> co <- colors()
> #-----
> # a dimensão do objeto
> length(co)
[1] 657
> #-----
> # a dimensão do objeto que contém apenas os elementos nas posições múltiplas de 7
> m7 <- seq(7, length(co), by=7) # sequencia multipla de 7
> m7
[1] 7 14 21 28 35 42 49 56 63 70 77 84 91 98 105 112 119 126 133
[20] 140 147 154 161 168 175 182 189 196 203 210 217 224 231 238 245 252 259 266
[39] 273 280 287 294 301 308 315 322 329 336 343 350 357 364 371 378 385 392 399
[58] 406 413 420 427 434 441 448 455 462 469 476 483 490 497 504 511 518 525 532
[77] 539 546 553 560 567 574 581 588 595 602 609 616 623 630 637 644 651
> length(co[m7])
[1] 93
> #-----
> # dimensão do objeto que contém apenas os elementos nas posições múltiplas de 7 e 11
> m77 <- seq(77, length(co), by=77) # multiplos de 7 e 11 são os multiplos de 77
> m77
[1] 77 154 231 308 385 462 539 616
> length(co[m77])
[1] 8
> #-----
> # a dimensão do objeto que contém apenas os elementos nas posições múltiplas de 7 ou 11
> m11 <- seq(11, length(co), by=11) # multiplos de 11
> m11
[1] 11 22 33 44 55 66 77 88 99 110 121 132 143 154 165 176 187 198 209
[20] 220 231 242 253 264 275 286 297 308 319 330 341 352 363 374 385 396 407 418
[39] 429 440 451 462 473 484 495 506 517 528 539 550 561 572 583 594 605 616 627
[58] 638 649
> m7ou11 <- c(m7,m11)
> sort(m7ou11) # alguns números são multiplos de ambos, ex 77, remover a intersecção

```

```

[1] 7 11 14 21 22 28 33 35 42 44 49 55 56 63 66 70 77 77
[19] 84 88 91 98 99 105 110 112 119 121 126 132 133 140 143 147 154 154
[37] 161 165 168 175 176 182 187 189 196 198 203 209 210 217 220 224 231 231
[55] 238 242 245 252 253 259 264 266 273 275 280 286 287 294 297 301 308 308
[73] 315 319 322 329 330 336 341 343 350 352 357 363 364 371 374 378 385 385
[91] 392 396 399 406 407 413 418 420 427 429 434 440 441 448 451 455 462 462
[109] 469 473 476 483 484 490 495 497 504 506 511 517 518 525 528 532 539 539
[127] 546 550 553 560 561 567 572 574 581 583 588 594 595 602 605 609 616 616
[145] 623 627 630 637 638 644 649 651

> m7ou11 <- unique(m7ou11) # unique remove os valores repetidos, deixa valores únicos
> length(co[m7ou11])

[1] 144

> # ou ainda, regra da adição de probs, P(A união B) = P(A)+P(B)-P(A inter B)
> length(m7)+length(m11)-length(m77)

[1] 144

> #-----
> # quantos elementos possuem a palavra "red" no nome
> red <- grep("red", co)
> length(co[red])

[1] 27

> # ou ainda
> length(grep("red", co))

[1] 27

> #-----
> # quantos elementos não possuem a palavra "blue" no nome
> blue <- grep("blue", co)
> length(co[-blue])

[1] 591

> # ou ainda
> length(co)-length(grep("blue", co))

[1] 591

> #-----
> NA

```

4. Importe os dados referentes à área dos municípios brasileiros no ano de 2000, obtenha o que se pede. Passe para o papel o código usado.

```

area <- read.table("http://www.leg.ufpr.br/~walmes/ensino/ce223-2011-01/area-mun-2000.txt",
                  header=TRUE, sep="\t", quote="")

```

- qual o número de municípios que possuem área registrada?
- qual a soma das áreas dos municípios em cada estado (no papel liste 5 deles)?
- qual a qual a área média dos municípios de cada estado (no papel liste 5 deles)?
- qual o valor da maior área municipal? a qual município pertence?
- qual a soma da área municipal das capitais do Paraná e Rio Grande do Sul?
- crie um novo objeto apenas com os municípios do Mato Grosso do Sul e liste os 5 maiores municípios e suas áreas.

```

> #-----
> # importa o arquivo de dados
> area <- read.table("http://www.leg.ufpr.br/~walmes/ensino/ce223-2011-01/area-mun-2000.txt",
+                   header=TRUE, sep="\t", quote="")
> #-----
> # municípios que possuem área registrada, então remover os NA
> area <- area[complete.cases(area),]
> str(area)

'data.frame':      5507 obs. of  3 variables:
 $ Sigla      : Factor w/ 27 levels "AC","AL","AM",...: 1 1 1 1 1 1 1 1 1 1 ...
 $ Município: Factor w/ 5310 levels "Abadia de Goiás",...: 27 396 706 750 1021 1412 1602 1677 2498 2768 ...
 $ area.km   : num  1603 2873 4339 3426 1717 ...

> nrow(area)

[1] 5507

> #-----
> # soma das áreas dos municípios em cada estado
> with(area, tapply(area.km, Sigla, sum))

```

```

      AC      AL      AM      AP      BA      CE      DF      ES
152521.9 27818.9 1570947.1 142815.8 564272.3 145712.3 5801.9 46047.2
      GO      MA      MG      MS      MT      PA      PB      PE
340118.8 331918.6 586553.0 357139.9 903385.2 1247703.1 56341.0 98525.7
      PI      PR      RJ      RN      RO      RR      RS      SC
251311.2 199282.1 43797.1 53077.3 237564.6 224118.1 268836.0 95286.1
      SE      SP      TO
21962.4 248176.6 277297.4
> #-----
> # área média dos municípios de cada estado
> with(area, tapply(area.km, Sigla, mean))
      AC      AL      AM      AP      BA      CE      DF
6932.8136 275.4347 25337.8565 8925.9875 1359.6923 791.9147 5801.9000
      ES      GO      MA      MG      MS      MT      PA
598.0156 1405.4496 1529.5788 687.6354 4638.1805 7169.7238 8725.1965
      PB      PE      PI      PR      RJ      RN      RO
252.6502 532.5714 1137.1548 499.4539 481.2868 319.7428 4568.5500
      RR      RS      SC      SE      SP      TO
14941.2067 575.6660 325.2085 292.8320 384.7699 1994.9453
> #-----
> # valor da maior área municipal e o município a qual pertence
> max(area$area.km)
[1] 160755
> area[which.max(area$area.km),]
      Sigla Município area.km
2436    PA  Altamira 160755
> #-----
> # soma da área municipal das capitais do Paraná e Rio Grande do Sul
> capi <- subset(area, Município%in%c("Curitiba","Porto Alegre"))
> sum(capi$area.km)
[1] 925.5
> # ou ainda
> sum(area$area.km[area$Município%in%c("Curitiba","Porto Alegre")])
[1] 925.5
> #-----
> # objeto com os municípios do MS e os 5 maiores municípios e suas áreas
> MS <- subset(area, Sigla=="MS")
> MS[order(MS$area.km, decreasing=TRUE),][1:5,]
      Sigla      Município area.km
2234    MS      Corumbá 64964.9
2270    MS      Porto Murtinho 17734.8
2271    MS      Ribas do Rio Pardo 17308.7
2217    MS      Aquidauana 16958.5
2209    MS      Água Clara 11031.0
> #-----

```

5. Acrescente a coluna do logaritmo da área dos municípios ao objeto e obtenha o que se pede. Passe para o papel o código usado.

- quantas classes tem o histograma dessa variável?
- refaça o histograma dessa variável com classes de amplitude 0,75.
- adicione uma linha vertical para representar a média dessa variável.
- quantos municípios possuem valor superior a essa média?
- faça o gráfico boxplot para essa variável em função dos estados.
- analise o gráfico e liste um estado com distribuição simétrica e um com assimetria à esquerda.

```

> #-----
> # acrescentando a coluna do log
> area$larea <- log(area$area.km)
> #-----
> # número de classes do histograma dessa variável
> ht <- hist(area$larea, freq=FALSE) # você pode contar o número de classes nos dedinhos
> str(ht)
List of 7
 $ breaks      : num [1:12] 1 2 3 4 5 6 7 8 9 10 ...
 $ counts      : int [1:11] 2 7 82 765 1838 1509 799 328 127 36 ...
 $ intensities: num [1:11] 0.000363 0.001271 0.01489 0.138914 0.333757 ...
 $ density     : num [1:11] 0.000363 0.001271 0.01489 0.138914 0.333757 ...
 $ mids       : num [1:11] 1.5 2.5 3.5 4.5 5.5 6.5 7.5 8.5 9.5 10.5 ...
 $ xname      : chr "area$larea"
 $ equidist   : logi TRUE
 - attr(*, "class")= chr "histogram"

```

```
> length(ht$counts) # você pode obter insecionado o objeto retornado pela função gráfica
```

```
[1] 11
```

```
> #-----
```

```
> # histograma com classes de amplitude 0,75
> hist(area$larea, freq=FALSE, breaks=seq(0,12,0.75))
```

```
> #-----
```

```
> # linha vertical para representar a média
> abline(v=mean(area$larea))
```

```
> #-----
```

```
> # municípios possuem valor superior a média
> sum(area$larea>mean(area$larea))
```

```
[1] 2440
```

```
> #-----
```

```
> # faça o gráfico boxplot para essa variável em função dos estados
> boxplot(larea~Sigla, data=area)
```

```
> #-----
```

```
> # liste um estado com distribuição simétrica e um com assimetria à esquerda
```

```
> abline(v=c(8,22), col="gray50", lty=2)
```

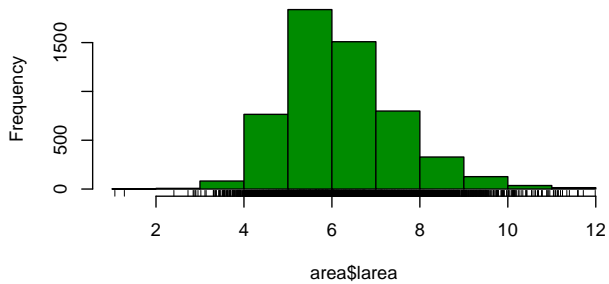
```
> text(c(8,22), 2, label=c("simetrico","ass esquerda"))
```

```
> levels(area$Sigla)[c(8,22)]
```

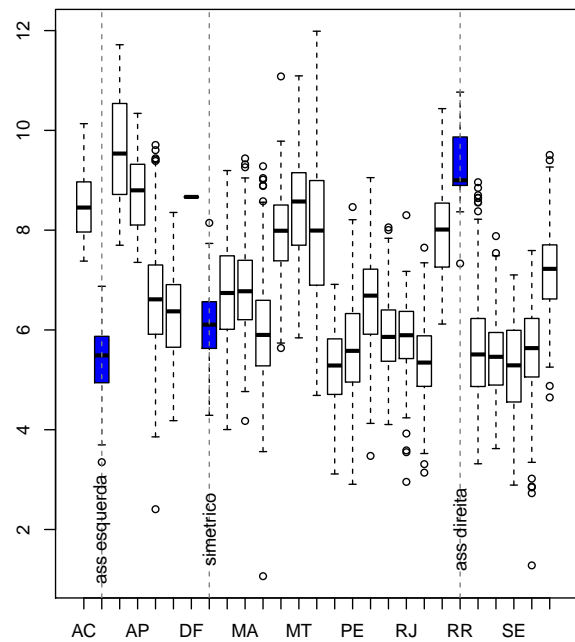
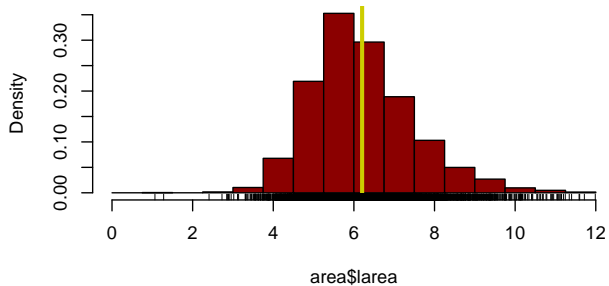
```
[1] "ES" "RR"
```

```
> #-----
```

Histogram of area\$larea



Histogram of area\$larea



Funções úteis: abline, apply, apropos, boxplot, c, class, complete.cases, curve, data.frame, density, dim, grep, help, hist, is.na, length, lines, log, ls, matrix, max, mean, median, min, ncol, nrow, order, plot, range, read.table, rev, rug, seq, sort, str, subset, sum, summary, tapply, unique, var, which.max, which.min, with .
