SIQUEIRA CAMPOS

Guia de referência R

www.siqueiracampos.com

12/02/2016 - Versão 1.0

1. Básicos

Comandos	`objects()`	Lista os objetos do workspace
	`ls()`	Lista os objetos do workspace
	`rm(obj)`	Exclui o objeto 'obj '
	`str(obj)`	Exibe a estrutura interna de 'obj', informando o tipo de objeto, número de observações e classe de cada variável.
Atribuições	`<-`	Atribui um valor à variável
	`=`	Atribui um valor à variável
Ajuda	`help(fun)`	Exibe o arquivo de ajuda da função fun()
	`?fun`	Exibe o arquivo de ajuda da função fun()
	`??palavra`	Pesquisa todas as funções que tenham a string “palavra” no nome ou descrição
	`args(fun)`	Lista os argumentos da função fun()
Bibliotecas e pacotes	`library(pact)`	Adiciona à memória as funcionalidades do pacote 'pact'
	`library(help=pact)`	Exibe a descrição do pacote 'pact'
	`install.packages("pact")`	Baixa o pacote 'pact' do repositório de pacotes do R

2. Vetores e tipos de dados

Geração	`seq(-3,3,0.1)`	Sequência: -3.0, -2.9, -2.8, ..., 2.9, 3.0
	`3:8`	Mesmo que seq(3,8,1)
	`c(4,6,8,1:3)`	Concatenação de vetores: 4 6 8 1 2 3
	`rep(2,5)`	2 2 2 2 2
	`rep(3:5,1:3)`	3 4 4 5 5 5
	`gl(3,2,12)`	Um fator com 3 níveis, repetindo cada nível em blocos de 2, até o comprimento 12. (1 1 2 2 3 3 1 1 2 2 3 3)
Coerção	`as.numeric(x)`	Converte para numérico
	`as.character(x)`	Converte para uma sequência de texto
	`as.logical(x)`	Converte para booleano
	`factor(x)`	Cria um fator com o vetor x
	`unlist(x)`	Converte uma lista ou um resultado de uma tabela para um vetor.

3. Data frames (quadro de dados)

Acesso aos dados	`data.frame(altura, peso)`	Cria um data frame com os vetores 'altura' e 'peso'.
	`infos$peso`	Seleciona o vetor 'peso' no data frame 'infos'
	`attach(infos)`	Coloca o data frame no caminho de busca, isso possibilita acessar as colunas do quadro diretamente pelos nomes.
	`detach(infos)`	Remove o data frame do caminho de busca.
Editando	`infos2 <- Edit(infos)`	Abre o data frame 'infos' no editor de dados, e escreve a versão alterada de 'infos' em 'infos2'
	`fix(infos)`	Abre o data frame 'infos' no editor de dados, as alterações serão feitas no próprio data frame 'infos'
Resumindo	`dim(infos)`	Dá o número de linhas e colunas do data frame 'infos', funciona também para vetores e matrizes
	`summary(infos)`	Da o resumo estatístico de cada variável do data frame 'infos'

Guia de referência R

4. Leitura e gravação de dados

Geral	`data(nome)`	Lê o data frame 'nome' do pacote datasets
	`read.table("arquivo.txt")`	Le um arquivo externo ASCII
Argumentos do read.table()	`header = TRUE`	Na primeira linha estão os nomes das variáveis
	`row.names = 1`	Na primeira coluna estão os nomes das colunas
	`sep = ","`	Os dados estão separados por vírgula
	`sep = "\t"`	Os dados estão separados por tab
	`dec = ","`	O separador decimal é a vírgula
	`na.strings = "."`	Valores faltantes são representados por ponto
Variações do read.table()	`read.csv("arquivo.csv")`	Vírgula como separador de dados
	`read.delim("arquivo.txt")`	Tab como separador de dados
Escrevendo	`write.table()`	Exporta o conjunto de dados
Adicionando nomes	`names()`	Nomeia colunas de um data frame ou de uma lista
	`dimnames()`	Nomeia linhas e colunas de uma matriz

5. Indexação / seleção / classificação

Vetores	`x[1]`	Primeiro Elemento
	`x[1:5]`	Subvetor contendo os primeiros cinco elementos
	`x[c(2,3,5)]`	Elementos de números 2, 3 e 5
	`x[y <= 25]`	Seleção por uma expressão lógica
	`x[sexo = = "feminino"]`	Seleção por variável fator
	`i <-c(2,3,5); x[i]`	Seleção por variável numérica
	`k <- (y <=25); x[k]`	Seleção por variável lógica
	`length(x)`	Retorna o tamanho do vetor
Matrizes e data frames	`m[4, ]`	Quarta linha
	`m[ ,3]`	Terceira coluna
	`dados[dados$var <=25, ]`	Parte de um data frame (não funciona para matrizes)
	`subset(dfr,var<=30)`	O mesmo mais simples (não funciona para matrizes)
	`m[m[ ,3]<=30, ]`	Parte de uma matriz (funciona para data frames)
Classificação	`sort(c(8,9,10,6))`	Retorna o vetor ordenado: 6, 8, 9, 10
	`order(c(11,9,10,6))`	Retorna os índices dos elementos quando ordenados em forma crescente: 4, 2, 3, 1
	`order(c(11,9,10,6), decreasing = TRUE)`	Faz o mesmo, mas ordenando os valores decrescentemente: 1, 3, 2, 4
	`rank(c(7,9,10,6))`	Retorna os índices dos elementos quando o vetor é posto em ordem de valores ascendentes: 2, 3, 4, 1

Guia de referência R

6. Valores faltantes

Funções	`is.na(x)`	Retorna um vetor lógico com TRUE onde x for NA
	`complete.cases(x1, x2,...)`	Retorna um vetor lógico indicando quais vetores não possuem valores faltantes
Argumentos para outras funções	`na.rm =`	Em funções estatísticas, se informa na.rm=TRUE caso haja valores em falta no argumento. Caso contrário o cálculo da função retorna NA.
	`na.last =`	Em 'sort' se na.last=TRUE, os valores faltantes ficam no final do ordenamento, se FALSE ficam no início e se NA são descartados.
	`na.action =`	Em 'lm', e outras funções, indica-se o que deve acontecer com os valores em falta 'na.fail', 'na.omit' ou 'na.exclude'
	`na.print =`	Em 'summary()' e 'print()' indica-se como representar os valores faltantes na saída de dados.
	`na.strings =`	Em 'read.table() ' indica-se o que serão considerados casos faltantes na leitura de dados

7. Funções numéricas

Matemáticas	`log(x)`	Logaritmo natural de x, ou seja, o logaritmo de base e
	`log(x, 10)`	Logaritmo de x na base 10
	`exp(x)`	Valor da função exponencial e^x
	`sin(x)`	Seno
	`cos(x)`	Cosseno
	`tan(x)`	Tangente
	`asin(x)`	Arco seno (Inverso do seno)
	`min(x)`	Menor valor do vetor
	`min(x1, x2, ...)`	Menor valor dentre vários vetores
	`max(x)`	Maior valor do vetor
	`range(x)`	O mesmo que c(min(x), max(x))
	`pmin(x1, x2, ...)`	Retorna um vetor com o mínimo paralelo (o mínimo indo elemento a elemento dos vetores)
	`length(x)`	Número de elementos do vetor
	`sum(x)`	Soma dos valores do vetor
	`cumsum(x)`	Soma cumulativa dos valores dos vetores
	`sum(complete.cases(x))`	Número de elementos não faltantes
Estatísticas	`mean(x)`	Média
	`median(x)`	Mediana
	`quantile(x, p)`	Quartis: mediana = quantile(x, 0.5)
	`var(x)`	Variância
	`sd(x)`	Desvio Padrão
	`cor(x, y)`	Correlação de Pearson
	`cor(x, y, method = "spearman")`	Correlação de postos de Spearman

8. Programação

Execução condicional	`if(p< 0.5)` `  print("Viva")`	Imprime "Viva" se a condição for verdadeira

	`if(p < 0.5) {`
	`  print("Viva")`
	`  i = i + 1`
	`}`
		Se a condição for verdadeira, todos os comandos dentro das chaves são executados { }

	`if(p < 0.5) {`
	`  print("Viva")`
	`} else {`
	`  i = i + 1`
	`}`
		Execução condicional com uma alternativa

Laços de repetição	`for(i in 1:10) {`
	`  print(i)`
	`}`
		Repete 10 vezes a mesma instrução

	`i <- 1`
	`while(i <= 10) {`
	`  print(i)`
	`  i = i + 1`
	`}`
		Faz o mesmo de forma mais complicada

Função definida pelo usuário	`fun <- function(a, b, fazer = FALSE {`
	`  if(fazer) {`
	`  a + b`
	`  }`
	`  else 0`
	`}`
		Define a função 'fun' que devolve a soma de 'a' com 'b' se o argumento 'fazer' for verdadeiro, ou zero se 'fazer' for falso.

9. Operadores

Aritméticos	`+`	Adição
	`-`	Subtração
	`*`	Multiplicação
	`/`	Divisão
	`^`	Elevar a potência
	`%/%`	Divisão inteira 5 %/% 3 = 1
	`%%`	Resto da divisão inteira: 5 %% 3 = 2
Lógicos ou relacionais	`==`	Igual
	`!=`	Diferente
	`<`	Menor
	`>`	Maior
	`<=`	Menor ou igual
	`>=`	Maior ou Igual
	`is.na(x)`	Valor faltante?
	`&`	Lógico E
	`|`	Lógico OU
	`!`	Lógico NÃO
	`which(a>10)`	Retorna os índices de 'a' que possuem conteúdo maior que 10

Guia de referência R

10. Tabulação, agrupamento e recodificação

Geral	`table(x)`	Cria uma tabela de frequência com os dados de x
	`table(x, y)`	Cria uma tabela cruzada com as informações de x e y
	`xtabs(~ x + y)`	Interface de fórmula para a tabela cruzada: utilize summary() para o teste do chi-square
	`factor(x)`	Converte vetor em fator
	`cut(x, breaks)`	Quebra uma variável contínua em subgrupos.
Argumentos de factor()	`levels = c()`	Os valores de x para codificar. Utilize se alguns valores não estão presentes nos dados, ou se a ordem estiver errada.
	`labels = c()`	Valores associados com os níveis (levels) do fator
	`exclude = c()`	Valores para excluir. O padrão é NA, defina como NULL para ter os valores em falta incluídos como um nível.
Argumentos do cut()	`breaks = c()`	São os pontos de quebra da variável contínua. Valores de breaks fora de x produzirão NA. Também pode ser um único número que será a quantidade de pontos de quebra na variável.
	`labels = c()`	Nomes dos grupos. O padrão é 1, 2, ...
Recodificando fatores	`levels(f) <- names`	Novo nome para níveis
	`factor(novcod [f])`	Combinando os níveis: 'novcode', por exemplo, c(1,1,1,2,3) junta os três primeiros dos 5 grupos do fator 'f'

11. Manipulação de matrizes e listas

Álgebra de matrizes	`m1 % * % m2`	Produto de matrizes
	`t(m)`	Matriz transposta
	`m[lower.tri(m)]`	Retorna os valores da matriz triangular inferior de m como um vetor
	`diag(m)`	Retorna os elementos da diagonal da matriz m
	`matrix(x, dim1, dim2)`	Utiliza o vetor x para preencher uma matriz com 'dim1' linhas e 'dim2' colunas
Operações complementa-res	`apply(m, dim, fun)`	Aplica a função 'fun' para cada linha (dim = 1) ou coluna (dim = 2) da matriz m
	`tapply(m, list(f1, f2), fun)`	Pode ser usado para agregar colunas ou linhas dentro da matriz m, conforme definido pela f1, f2, usando a função 'fun' (por exemplo mean ou max)
	`split(x, f)`	Separa o vetor, a matriz ou o data frame 'f' pelo fator 'x'. Produz um resultado diferente para matriz e para data frame! O resultado é uma lista com um objeto para cada nível de 'f'.
	`sapply(list, fun)` `sapply(split(x,f), fun)`	Aplica a função 'fun' para cada objeto de uma lista, por exemplo como a criada através da função split

Guia de referência R

12. Métodos estatísticos padrão

Testes paramétricos, dados contínuos	`t.test`	Teste t para uma e duas amostras
	`pairwise.t.test`	Teste de média para dados pareados
	`var.test`	Teste de duas variâncias (F-test)
	`lm(y ~ x)`	Análise de Regressão
	`lm(y ~ f)`	Análise de Variância Simples (One-way)
	`lm(y ~ x1 + x2 + x3)`	Regressão Múltipla
	`lm(y ~ f1 * f2)`	Análise de variância 2 fatores (Two-way)
Não paramétricos	`wilcox.test`	Teste Wilcox para uma e duas amostras
	`kruskal.test`	Teste Kruskal-Wallis
	`friedman.test`	Análise de variância de Friedman (Two-way)
Variante de cor.test	`method = "spearman"`	Correlação de postos de Spearman
Resposta discreta	`binom.test`	Teste binomial (inclui teste do sinal)
	`prop.test`	Teste de proporções
	`fisher.test`	Teste exato em tabelas 2 x 2
	`chisq.test`	Teste qui-quadrado de independência
	`glm(y ~ x1+x2, binomial)`	Regressão Logística

13. Distribuições estatísticas

Distribuição normal	`dnorm(x)`	Função densidade
	`pnorm(x)`	Função de distribuição acumulada P(X<=x)
	`qnorm(p)`	Função quantil, retorna x em: P(X<=x) = p
	`rnorm(n)`	n números aleatórios normalmente distribuídos
Distribuições	`pnorm(x, mean, sd)`	Normal
	`plnorm*x, mean, sd)`	Lognormal
	`pt(x, df)`	t student
	`pf(x, n1, n2)`	F
	`pchisq(x, df)`	Qui-quadrado
	`pbinom(x, n, p)`	Binomial
	`ppois(x, lambda)`	Poisson
	`punif(x, min, max)`	Uniforme
	`pexp(x, rate)`	Exponencial
	`pgamma(x, shape, scale)`	Gama
	`pbeta(x, a, b)`	Beta

14. Modelos

Fórmulas de modelos	~	Como explicado por
	+	Efeitos Aditivos
	:	Interação
	*	Efeitos principais + interação: a*b = a+b+a:b
	-1	Remove intercepto
Modelos lineares	`lm.out <- lm(y ~ x)`	Ajusta o modelo e salva o resultado como 'lm.out'
	`summary(lm.out)`	Coeficientes etc.
	`anova(lm.out)`	Tabela da analise de variância
	`fitted(lm.out)`	Valores ajustados
	`resid(lm.out)`	Resíduos
	`predict(lm.out, newdata)`	Previsões para um novo data frame
Outros modelos	`glm(y ~ x, binomial)`	Regressão Logística
	`glm(y ~ x, poisson)`	Regressão de Poisson
	`gam(y ~ s(x))`	Modelo aditivo geral para regressão não-linear com suavização. Pacote: Gam
	`tree(y ~ x1+x2+x3)`	Classificação (y = fator) ou regressão (y = numérico). Pacote: tree
Diagnósticos	`rstudent(lm.out)`	Resíduos Studentizados
	`dfbetas(lm.out)`	Mudança na regressão padrão, coeficientes beta se observação removida.
	`dffits(lm.out)`	Mudanças no ajuste se observação removida
Análise de sobrevivência	`S <- Surv(time,ev)`	Cria objeto sobrevivência. Pacote: survival
	`survfit(S)`	Estimativa de Kaplan-Meier
	`plot(survfit(S))`	Curva de sobrevivência
	`survdiff(S ~ g)`	(Log-rank) teste para igualdade de curvas de sobrevivência
	`coxph(S ~ x1 + x2)`	Modelo de risco proporcional de Cox's

Multivariada	`dist()`	Calcula distância, Euclidiana ou outras
	`hclust()`	Análise Cluster hierárquico
	`kmeans()`	Análise de Cluster k-means
	`rda()`	Realiza análise de componente principal PCA ou análise de redundância. Pacote 'vegan'.
	`cca()`	Realiza (canônica) análise de correspondência, CA /CCA. Pacote: 'vegan '
	`diversity()`	Calculo de índices de diversidade. Pacote: 'vegan '

15. Gráficos

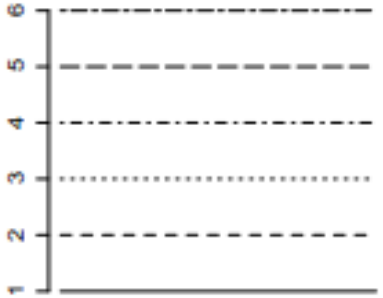
Gráficos padrão	`plot(x, y)`	Dispersão (ou outro tipo de gráfico se x e y não são vetores numéricos)
	`plot(f, y)`	Conjunto de boxplots para cada nível do fator 'f'
	`hist()`	Histograma
	`boxplot()`	Boxplot
	`barplot()`	Barra
	`dotplot()`	Dot plot
	`piechart()`	Pizza
	`interaction.plot()`	Gráfico de interação (Análise de variância)
Desenhando elementos (adicionando ao gráfico)	`lines()`	Linhas
	`abline()`	Linha de regressão
	`points()`	Pontos
	`arrows()`	Setas (obs: angle = 90 gera erro)
	`box()`	Moldura do gráfico
	`title()`	Título (acima do gráfico)
	`text()`	Texto no gráfico
	`mtext()`	Texto da margem
	`legend()`	Lista de símbolos
Parâmetros gráficos: argumentos de par()	`pch`	Símbolos (veja ao lado)
	`mfrow, mfcol`	Painel com múltiplos gráficos
	`xlim, ylim`	Limites do gráfico
	`lty, lwd`	Tipo de linha / largura (veja ao lado)
	`col`	Cor das linhas ou símbolos (veja ao lado)

Símbolos dos pontos (pch)

Cores (col)

1 - preto
2 - vermelho
3 - verde
4 - azul
5 - azul claro
6 - roxo
7 - amarelo
8 - cinza

Tipos de linha (lty)



Fontes: Modificado do livro: P. Dalgaard (2002). Introductory Statistics with R. Springer, New York.

R Development Core Team. R: A Language and Environment for Statistical Computing.

Vienna, Austria, 2016. Disponível em: http://www.R-project.org/