

Linguagem de Programação S Fundamentos e Aplicações em Recursos Florestais



5. Noções de Programação

5.5. Exemplos de Algumas Funções

5.5.1. Uma função "Jackknife"

Código da Função

```
1 #####
2 ##### 5. NOÇÕES DE PROGRAMAÇÃO
3 ##### 5.5. Exemplo de Algumas Funções
4 #####
5 # 5.5.1. Uma função "Jackknife"
6 ## Código da Função
7 jackknife <- function(x, FUN, ...)
8 {
9     n = length(x)
10    X = matrix(rep(x, n), ncol=n, byrow=F)
11 #
12    diag(X) = rep(NA, n)
13 #
14    out = apply(X, 2, FUN, ...)
15 #
16    return(out)
17 }
18
```

Exemplo de Aplicação

Dados: [caixeta.csv](#)

```
19 ## Exemplo de Aplicação
20
21 cax = read.csv("caixeta.csv", as.is=TRUE)
```

```
22 cax$dap = cax$cap / (10*pi)
23 d.mediana = jackknife(cax$dap, median, na.rm=T)
24 plot(density(d.mediana))
25 abline(v=median(cax$dap), col="red")
26
```

5.5.2. Cálculos da "Species Accumulation Curve"

Código da Função

```
27 # 5.5.2. Cálculos da "Species Accumulation Curve"
28 ## Código da Função
29 sac.curve <- function(x, index=1:(dim(x)[1]))
30 {
31 #     x -> matriz de presenca- ausencia das especies
32 #         linhas = unidades amostrais (parcelas, pontos, etc.)
33 #         colunas = especies
34 #     index -> vetor de indices para as linhas de x
35 #
36 #
37     x <- x[index,]
38     n.ua <- dim(x)[1]
39     n.sp <- dim(x)[2]
40     acumula <- apply(x, 2, cumsum)
41     difere <- apply(acumula, 2, diff)
42     reverte <- acumula[ - n.ua, ] == 0
43     coletor <- rbind( x[1,], difere * reverte)
44     dimnames(coletor)[1] <- dimnames(x)[1]
45     out <- cumsum( apply(coletor, 1, sum) )
46     out
47 }
48
```

Exemplo de Aplicação

Dados: [caixeta.csv](#)

```
49 ## Exemplo de Aplicação
50
51 cax = read.csv("caixeta.csv")
52 cax$arv = paste(cax$local, cax$parcela, cax$arvore)
53
54 jureia = cax[ cax$local == "jureia", ]
55 retiro = cax[ cax$local == "retiro", ]
56 chaus = cax[ cax$local == "chaus", ]
```

```

57
58 jureia.sac = sac.curve( table( jureia$arv, jureia$especie ) )
59 retiro.sac = sac.curve( table( retiro$arv, retiro$especie ) )
60 chauas.sac = sac.curve( table( chauas$arv, chauas$especie ) )
61
62 plot( 1:length(jureia.sac),
63       jureia.sac, xlab="Número de Árvores", ylab="Número de Espécies",
64       type = "l", col="red" , xlim=c(1,290), lwd=3)
65 lines( 1:length(retiro.sac), retiro.sac, col="green", lwd=3)
66 lines( 1:length(chauas.sac), chauas.sac, col="orange", lwd=3)
67 legend(150, 13, c("Juréia","Chauás","Retiro"), lwd=3,
68        col=c("red","green","orange") )
69

```

5.5.3. Uma função "plot" para Objetos "nls"

Código da Função

```

70 # 10.5.3. Uma função "plot" para Objetos "nls"
71 ## Código da Função
72 plot.nls <- function(x, which=c(1:4), ...)
73 {
74     old.par <- par(no.readonly = TRUE)
75     on.exit(par(old.par))
76     par( ask = TRUE )
77 #
78     if( !any(class(x) == "nls") )
79         stop("Only work with nls objects!")
80     if( any(which==1) )
81     {
82         plot(fitted(x), residuals(x),
83              xlab = "Fitted Values",
84              ylab = "Residuals",
85              ...
86             )
87         abline( h = 0, lty=2 )
88         lines(lowess(fitted(x), residuals(x)), col="red")
89     }
90     if( any(which==2) )
91     {
92         qqnorm( residuals(x),
93                ylab = "Residual Quantiles",
94                ...
95               )
96         qqline( residuals(x), lty=2 )
97         #
98     }

```

```
99     if( any(which==3) )
100     {
101         xval = fitted(x)
102         yval = sqrt(abs(residuals(x)))
103         plot(xval, yval,
104              xlab = "Fitted Values",
105              ylab = "Sqrt( Abs( Residuals ) )",
106              ...
107            )
108         lines(lowess(xval, yval), col="red")
109     }
110     if( any(which==4) )
111     {
112         prof <- profile(x)
113         plot( prof, absVal=FALSE )
114     }
115 #
116     invisible()
117 }
118
```

Exemplo de Aplicação

Dados: [caixeta.csv](#)

```
119 ## Exemplo de Aplicação
120 coef( lm(log(h) ~ log(cap), data=cax) )
121 nlin = nls( h ~ exp(b0)*cap^b1, data=cax, start=c(b0=2.5316, b1=0.34028)
122 )
123 plot(nlin)
```

5.5.4. Cálculo dos "Variance Inflation Factors (VIFs)"

Código da Função

```
124 # 5.5.4. Cálculo dos "Variance Inflation Factors (VIFs)"
125 ## Código da Função
126 vif <- function(object, ...)
127     UseMethod("vif")
128
129 vif.default <- function(object, ...)
130     stop("No default method for VIF. Sorry.")
131
132 vif.lm <- function(object, ...)
```

```
133 {
134     V <- summary(object)$cov.unscaled
135     Vi <- crossprod(model.matrix(object))
136     nam <- names(coef(object))
137     k <- match("(Intercept)", nam, nomatch=FALSE)
138     if( k )
139     {
140         v1 = diag(V)[-k]
141         v2 = diag(Vi)[-k] - Vi[k, -k]^2 / Vi[k, k]
142         nam = nam[-k]
143     }
144     else
145     {
146         v1 <- diag(V)
147         v2 <- diag(Vi)
148     }
149     structure(v1 * v2, names=nam)
150 }
151
```

Exemplo de Aplicação

Dados: [esaligna.csv](#)

```
152 ## Exemplo de Aplicação
153
154 esa = read.csv("esaligna.csv")
155 plot(total ~ dap, data=esa)
156 reg = lm( total ~ dap + I(dap^2) + ht + I(dap^2*ht), data=esa)
157 vif(reg)
158 vif(nlin)
159
```

From:

<http://insilvaarbores.com.br/dokuwiki/> - **In Silva, Arbores ...**

Permanent link:

http://insilvaarbores.com.br/dokuwiki/doku.php?id=pt:cursos_online:s_linguagem:5-nocoas-programacao:5-5-exemplo-funcoes

Last update: 2024/10/22 00:13

