CICLOS

Miranda Belmonte Hairo

9 de septiembre de 2016

Ejercicio 1

speed<-90  
while(speed >=30){  
 print("B("jale")  
 speed<-speed -7  
}  
speed

si voy a m("s de treinta, indicame que le baje, y mas aparte bajale 7 a la velocidad.

Fin Ejercicio 1

Ejercicio 2

speed<-64  
while(speed>48){  
 print(paste("Tu velocidad es",speed<-speed-11))  
if (speed>48){  
 print("B("jale pero mucho!")  
}else{  
 speed<-speed-6  
 print("B("jale")  
}  
}  
speed

Fin Ejercicio 2

Ejercicio 3

speed<-86  
while(speed>80){  
 print(paste("Tu velocidad es",speed<-speed-11))  
 if (speed>80){  
 break  
 }else{  
 speed<-speed-6  
 print("B("jale")  
 }  
}  
speed

Fin Ejercicio 3

Ejercicio 4

speed<-80  
while(speed>80){  
 print(paste("Tu velocidad es",speed<-speed-11))  
 if (speed>80){  
 break  
 }else{  
 speed<-speed-6  
 print("B("jale")  
 }  
}  
speed

Fin Ejercicio 4

i<-2  
  
while (i<10){  
 print(paste(i\*3,i+1))  
 if (i%%8==2){  
 break  
 }else{  
 print(i\*3)  
 }  
}

CICLO "FOR""

for(i in 1:10){  
 print(i)  
}  
i  
  
# si cambi(. i  
i<-54  
for (i in 1:10) {  
 print(i)  
 }  
i

INDEXANDO primos<-c(2,3,5,7,11,13)

versi(.n 1

for (p in primos){  
 print(p)  
}

versi(.n 2

for (i in 1:length(primos)){  
 print(primos[i])  
}

versi(.n 3

for (i in seq\_along(primos)){  
 print(primos[i])  
}

EJERCICIO 5 Considera el vector

linkedin<-c(16,9,13,5,2,17,14)

Escribe un ciclo for en cada de las tres versiones que imprima cadaelementode lindekin porseparado. Hacer un ciclo por una lista es similar, por ejemplo

for (l in linkedin){  
 print(l)  
}

for (i in 1:length(linkedin)){  
print (linkedin[i])  
}

for (i in seq\_along(linkedin)){  
 print(linkedin[i])  
}

Fin de Ejercicio

EJERCICIO 6 Considera la lista

nyc<- list(pop=8405837,  
suburbios = c("Manhattan","Bronx","Brooklyn","Queens","Staten Island"),  
 capital=FALSE)

Imprime sus elementos de las tres formas

Versi(.n 1

for (p in pop){  
 print(p)  
}

Versi(.n 2

for (t in 1:length(pop)){  
 print(pop[t])  
}

Versi(.n 3

for (t in seq\_along(pop)){  
 print(pop[t])  
}

Versi(.n 1

for ( s in suburbios){  
 print(s)  
}

Versi(.n 2

for (s in 1:length(suburbios)){  
 print(suburbios[s])  
}

Versi(.n 3

for (s in seq\_along(suburbios)){  
 print(suburbios[s])  
}

EJERCICIOS 7 En este ejercicio, la matriz gato representa el estado de un juego de gato. Marca con una !0X!1 y una !0O!1 donde hay estas jugadas y con !0NA!1 donde todav(\*a no se ha jugado. De023ne esta matriz en tu ("rea de trabajo

g<-c("0",NA,"X",NA,"0","0","X",NA,"X")  
gato<-matrix(g,nrow = 3,ncol = 3,byrow = TRUE)

Ejercicio 7

for (i in row(gato)) {  
 for (j in col(gato)) {  
 print(paste("En el renglon",i,"columnda",j,"el juego contiene",gato[i,j]))  
 }   
}

EJERCICIO 8

linkedin<-c(16,9,13,5,2,17,14)

for (i in linkedin) {  
 if(i>9){  
 print(paste("Popular",i))  
 }else{  
 print(paste("Hazte notar",i))  
 }  
}

EJERCICIO 9

for (i in linkedin){  
 if(i>9){  
 print(paste("Popular",i))  
 }else if (i>16){  
 print(paste("Ya es mucho",break))  
 }else if (i<5){  
 print(paste("Vergonzoso",next))  
 }else {  
 print(paste("Hazte notar",i))  
 }  
 }

EJERCICIO 10

rquote<- "los caminos internos de r son irrefutablemente intrigantes"  
chars <- strsplit(rquote, split = "")[[1]]  
cuenta.r=0  
for (i in chars) {  
 if (i=="r") {  
 print(cuenta.r<-cuenta.r+1)  
 }else if (i=="u") {  
 break  
 }   
}  
  
cuenta.r

lapply() Pasar un vector character de mayusculas a minusculas con la función tolower

?lapply  
nombres<-c("GAUSS","BERNOULLI","KOLMOGOROV")

Versión For

for (i in seq\_along(nombres)){  
 print(tolower(nombres[i]))   
}  
  
for (i in nombres) {  
 print(tolower(i))  
}

Versión lapply

minusculas <- lapply(nombres, tolower)   
minusculas   
str(minusculas)

select\_prim<-function(x){  
 x[1]  
}  
grandes<-c("GAUSS:1777","BERNOULLI:1700","KOLMOGOROV:1903")   
split<-strsplit(grandes, split=":") split\_min<- lapply(split, tolower)  
nombres<-lapply(split\_min, select\_prim)  
  
select\_sec<-function(y){  
 y[2]  
}  
años<-lapply(split\_min, select\_sec)

Funciones anónimas funciones dentro del propio lapply.

#Versión con Función pre-definida  
triple<-function(x){3\*x}  
lapply(list(1,2,3), triple)  
  
#Versión con Función anónima  
lapply(list(1,2,3), function(x){3\*x})

Ejercicio Usa lapply anónimo para el ejercicio pasado.

grandes<-c("GAUSS:1777","BERNOULLI:1700","KOLMOGOROV:1903")   
split<-strsplit(grandes, split=":")   
split\_min<- lapply(split, tolower)  
nombres<-lapply(split\_min,function(x){x[1]})  
años<-lapply(split\_min,function(y){y[2]})

lapply(), permite pasar otros argumentos extras.

multi<-function(x,factor){  
 x\*factor  
}  
  
multi(3,4)  
lapply(list(1,2,3),multi, factor=4)

EJERCICIO

select\_ind<-function(x,index){  
 x[index]  
}  
lapply(split\_min, select\_ind)

sapply() similar a lapply, primer argumento un vector o lista, segundo argumento una función pero con argumentos potenciales. s de sapply significa simplificación. Trata de simplificar a lapply. sapply llama a lapply, después simplifica.

temp <- list( c(3,6,9,7,-3), c(6,12,13,4,9), c(4,-1,-3,8,7), c(1, 7, 2,3, 4), c(5,9,8,4,5,-1), c(-3,0,3,-1,5,6), c(3,6,9,12,4,8) )

EJERCICIO

for (i in 1:7){  
 print(min(temp[[i]]))  
}  
va<-function(x){  
 min(x)  
}  
lapply(temp,va)  
sapply(temp,va)  
  
ve<-function(x){  
 max(x)  
}  
lapply(temp,ve)  
sapply(temp,ve)

EJERCICIO

prom\_extremos<-function(x){  
 min(x)/2  
}  
lapply(temp,prom\_extremos)  
sapply(temp,prom\_extremos)  
  
prom\_extremos2<-function(y){  
 max(y)/2  
}  
lapply(temp,prom\_extremos2)  
sapply(temp,prom\_extremos2)  
  
prom\_extremos3<-function(x){  
 c(max(x)/2,min(x)/2)  
}  
lapply(temp,prom\_extremos3)  
sapply(temp,prom\_extremos3)

EJERCICIO Utiliza extremos definidos.

extremos<- function(x){  
 c(min=min(x),max=max(x))  
}  
lapply(temp,extremos)  
sapply(temp,extremos)