Patricia Tachinardi

Mestranda do Departamento de Fisiologia do IB-USP, sob orientação da Profa. Gisele Akemi Oda, na área de cronobiologia. Estudo o ritmo de temperatura corporal em tuco-tucos (*Ctenomys* cf. *knighti*), roedores subterrâneos que habitam o noroeste da Argentina.

Meus Exercícios

exec

Proposta de Trabalho Final

Principal

Apresentação do problema

Actograma é uma forma de representação gráfica muito comum na cronobiologia. É uma ferramenta poderosa para a análise visual de ritmos. Sua inspeção é importante para verificar se a variável medida (atividade locomotora, temperatura corporal, etc.) tem ou não expressão rítmica e, se sim, quais são os padrões que esse ritmo exibe.

O conceito por trás de um actograma é simples. É como se um grande gráfico da série temporal fosse cortado a cada 24 horas e esses gráficos fossem empilhados, de forma que cada linha do novo gráfico corresponda a um dia. É comum plotar actogramas em duplicata (*double plot*), colocando dois gráficos idênticos um ao lado do outro de forma que se possa ver 48 horas em uma única linha, o que facilita muito a visualização de ritmos com períodos diferentes de 24 horas. A figura abaixo mostra o resultado final desse "empilhamento", na qual a variável medida é a atividade em roda de um tuco-tuco (as barras pretas indicam os momentos que o animal estava ativo). Uma inspeção visual dessa figura nos revela que a atividade em roda desse roedor é rítmica em condições constantes, com um período maior de 24 horas, e sincroniza-se com o ciclo de claro/escuro, quando o animal fica ativo durante a fase de escuro

Objetivo da proposta: Fazer uma função no R que retorne esse tipo de gráfico.

Plano de execução:

Entrada de dados: O formato dos dados a serem utilizados para a construção do actograma pode ser na forma de data frame ou matriz (acho que dependendo dos comandos usados um ou outro pode ser melhor...). Eles devem conter uma coluna com a data, uma com a hora e outra com as medidas da variável. É essencial que os intervalos sejam regulares.

O que a função deve retornar:

É essencial que:

1. Ela faça gráficos para cada dia da série temporal e depois os empilhe.
2. Ela permita que esses gráficos sejam plotados em duplicata (*double plot*)
3. Os horários constem no eixo x, e os dias no eixo y.
4. Ela permita que sejam selecionados valores máximos e mínimos a serem plotados. Ou seja, caso os dados sejam de temperatura corporal, pode-se selecionar, por exemplo, que apenas os valores entre 36 e 38 graus sejam plotados. Nos horários em que a temperatura não está entre esses valores o gráfico ficaria em branco. Isso é importante porque quando o actograma representa muitos dias é quase impossível observar a amplitude do ritmo, pois as linhas ficam muito pequenas. Deixar os espaços em branco, portanto, facilita muito a visualização

Além dessas características básicas, outras funcionalidades facilitariam muito a visualização dos dados e confecção de actogramas para publicação:

1. A possibilidade de selecionar, dentro de um data frame ou matriz com muitos dias, quais são aqueles que devem ser plotados no actograma, o que economizaria a criação de objetos intermediários.
2. A plotagem dos regimes de iluminação no próprio actograma. Esse tipo de representação facilita muito a interpretação dos dados, especialmente em actogramas muito extensos.
3. Um eixo indicando os regimes de iluminação ou outras condições (como diferentes temperaturas ambientes, por exemplo).

Comentários da Proposta Principal

A proposta está ótima. Você tem definido seu input e output, agora pense em cada passo do processo. Acho que a função sem as funcionalidades extras já está de bom tamanho. Legal o argumento que define os máximos e mínimos da variável plotada.

Gabriel

Plano B

Apresentação do problema

Para identificar a forma da onda de um ritmo é utilizado um gráfico denominado plexograma. É uma forma de representação gráfica na qual são plotadas as médias dos valores obtidos em todos os dias, por horário de medição. Dessa forma, o eixo x tem 24 horas e o y os valores da variável medida. Além de mostrar a forma da onda do ritmo, esse tipo de representação pode ser útil para mostrar relações de fase entre ritmos diferentes, quando são plotados no mesmo gráfico. A figura abaixo mostra um exemplo de medidas de atividade em roda e temperatura corporal feitas em um tuco-tuco. A barra superior mostra o regime de luz, sendo o claro representado pelo branco e o escuro pelo preto.

Objetivo da proposta: Criar uma função no R que faça as médias da variável por horário e construa com elas esse tipo de gráfico.

Plano de execução

Entrada de dados: O formato dos dados deve ser o mesmo descrito na proposta principal.

O que a função deve retornar:

Uma representação gráfica como a descrita acima, que permita, de preferência:

1. A plotagem de mais de uma variável.
2. Que seja colocada uma linha horizontal para indicar a média e/ou mediana de todos os dados.
3. Que sejam colocadas barras de desvio padrão para cada ponto
4. Que seja colocada uma barra com os regimes de iluminação.

Além disso , a função poderia retornar uma tabela com valores numéricos como mediana, média e desvio padrão totais, valores máximo e mínimos e médias máximas e mínimas.

Comentários

Essa idéia é está boa e não é difícil de executar.

Gabriel

Comentário sobre as funções

Eu esbarrei em alguns problemas com a minha proposta principal. Por causa da estratégia que escolhi para empilhar os gráficos, usando layout(),só é possível plotar gráficos correspondentes a no máximo 50 dias. Também por essa razão há pouca flexibilidade para mudar o eixo y.

Terminei essa função, mas como não sabia se ia conseguir, acabei fazendo o plano B também. Pretendo aperfeiçoar ambas para analisar meus dados do mestrado,portanto qualquer sugestão (principalmente em relação a como contornar os problemas da proposta principal) é mais do que bem-vinda.

Após a entrega

Você avançou muito para o pouco tempo que teve para fazer as funções, muito bom.

Não temos ninguém na equipe que conheça estas análises, mas se o seu foco não desenvolvimento de rotinas, eu buscaria pacotes no R já implementados, para você não começar do zero. Certamente estas funções que você fez irão te ajudar, mas a coisa parece complexa o suficiente para você aproveitar soluções encontradas por outras pessoas.

Função da Proposta Principal

Página de Ajuda

```
actogram                package:nenhum                R Documentation

Plotagem de Actogramas

Description:

     Produz um gráfico do tipo actograma a partir de séries temporais com
intervalos regulares. O gráfico será construído sobre um eixo x de 24 horas.
     A plotagem pode ser simples (singleplot) ou em duplicata (doubleplot).
Usage:

     actogram(x, intervalo=5, doubleplot=TRUE, lim.max="n", lim.min="n",
titulo=NULL, cex=1)

Arguments:

     x: Vetor numérico. Valores de uma variável medida em intervalos
regulares.
     intervalo: Numérico. Valor do intervalo entre medidas, em minutos.
     doubleplot: Lógico. Define se a plotagem será em duplicata
(doubleplot=TRUE) ou simples(doubleplot=FALSE)
     lim.max: Numérico. Valores máximos a serem plotados. Qualquer valor
maior que o selecionado não aparecerá no gráfico.
     Se lim.max="n" (dafault), o ajuste dos valores que aparecerão no
gráfico é automático, para cada dia. Essa opção pode gerar problemas, ver
warnings.
     lim.min: Numérico. Valores mínimos a serem plotados. Qualquer valor
menor que o selecionado não aparecerá no gráfico.
     Se lim.max="n" (dafault), o ajuste dos  valores que aparecerão no
gráfico é automático, para cada dia. Essa opção pode gerar problemas, ver
warnings.
     titulo: Caracter. Título que aparecerá no topo do actograma. Se
titulo=NULL, nenhum título aparecerá.
     cex: Numérico. Indica o quanto o tamanho do texto do título e dos eixos
deve ser aumento em relação ao default. Para mais detalhes ver o help da
função par.

Details:

     Cria um gráfico do tipo actograma, em que gráficos para cada dia da
série temporal são empilhados, de forma que cada linha
     do novo gráfico corresponda a um dia. É possível plotar o actogramas em
duplicata (doubleplot),de forma que se possa ver 48 horas
     em uma única linha. Os objeto deve ser um vetor numérico,
```

```
correspondentes a medidas tomadas com intervalos regulares e iniciando à
meia-noite.

Value:

     Um gráfico do tipo actograma é criado no dispositivo gráfico.

Warning:

     Para que a escala do eixo x corresponda aos horários reais, é
importante que as medidas que contam nos vetores iniciem à meia-noite.

     Esta função não aceita vetores com valores referentes a mais de 50
dias.

     Em alguns casos, utilizar os argumentos lim.min="n" e lim.max="n" pode
gerar erros. Caso isso aconteça, insira valores numéricos nos argumentos.

Author(s):

     Patricia Tachinardi Andrade Silva
     p.tachinardi@gmail.com

References:

     Moore-Ede M, Sulzman FM, Fuller CA. (1984). The Clocks That Time Us:
Physiology of the Circadian Timing System.
     Commonwealth Fund Publications. Harvard University Press.446 p.

See Also:

     plot() para gráficos genéricos
Examples:

     ## Dados fictícios, simulando dados obtidos em intervalos de 5 minutos
por 15 dias
     exemplo= c(rep(c(runif(144, 20, 40), runif(144, 0, 10)), 15))
     actogram(exemplo, titulo="Exemplo de Actograma", cex=1.1)
     ## Dados ficitícios, dimulando dados obtidos em intervalos de 15
minutos por 30 dias
     exemplo2= c(rep(c(runif(48, 30, 80), runif(48, 10, 50)), 30))
     actogram(exemplo2, titulo="Exemplo de Actograma 2", intervalo=15)
     # Agora com limites máximos e mínimos
     actogram(exemplo2, titulo="Exemplo de Actograma 2", intervalo=15,
lim.min=40, lim.max=80)

     ## Exemplo com dados reais
     # O arquivo "tuco.txt" contem dados da temperatura corporal de um tuco-
tuco por 42 dias, com intervalo entre medidas de 5 minutos
     tuco= read.table("tuco.txt")
     actogram(tuco, titulo="Temperatura corporal", lim.min=36.6,
```

```
lim.max=37.6, cex=0.8)
```

Código da Função

```
actogram = function(x, intervalo=5, doubleplot=TRUE, lim.max="n",
lim.min="n", titulo=NULL, cex=1)
{
med.dia=1440/intervalo # Numero de medidas por dia
if((length(x)/med.dia)>50)
    {
    cat("\n Erro: o número de dias não pode ultrapassar 50. \n")
    }
else
{
x11()
if(doubleplot==FALSE)
    {
    unico=matrix(nrow=med.dia, ncol=(length(x)/med.dia))
    unico[,1]=x[1:med.dia]
    for(i in 1:(length(ameg)/med.dia-1)) # Cortando os dados em dias
        {
        unico[,i+1]= x[((med.dia*i)+1):((med.dia*i)+med.dia)]
        }
    matriz=matrix(c(1:(ncol(unico))), (ncol(unico)))
    layout(matriz, heights=rep(2, nrow(matriz)))
    par(mar=c(0,0,0,0), bty="n", lend="butt", oma=c(4,3,4,1), las=1,
cex=cex)
    if(lim.max=="n"|lim.min=="n")
        {
        for(i in 1:ncol(unico))
            {
            plot(unico[,i], type="h", lwd=5, xaxt="n", yaxt="n", ylim=
c(lim.min, lim.max))
            axis(2, at=(c((max(unico[,i])), min(unico[,i]))),  labels=(c("
", "")))
            axis(2, at=( (max(unico[,i]) - min(unico[,i]))/2 +
min(unico[,i])), labels=(i), lty="blank")
            }
        }
    else
        {
        for(i in 1:ncol(unico))
            {
            plot(unico[,i], type="h", lwd=5, xaxt="n", yaxt="n",
ylim=c(lim.min, lim.max))
            axis(2, at=(c(lim.min, lim.max)),  labels=(c(" ", "")))
            axis(2, at=(((lim.max - lim.min)/2)+lim.min), labels=(i),
lty="blank")
            }
```

```
        }
    axis(1, at=seq(0, med.dia, (60/intervalo*3)), outer=TRUE, labels=seq(0,
24, 3), line=0.5, cex=0.8, xlab="Horas")
    title(main=titulo,  xlab="Horas", outer=TRUE, ylab="Dias")
    }
if(doubleplot==TRUE)
    {
    duplo=matrix(nrow=med.dia*2, ncol=((length(x)/med.dia)+1))
    dia1=x[1:med.dia]
    duplo[,1]= c(rep(NA, med.dia), dia1)
    dia2=x[((med.dia)+1):((med.dia)+med.dia)]
    duplo[,2]=c(dia1, dia2)
    dia = matrix(nrow=med.dia, ncol=(length(x)/med.dia))
    dia.x = matrix(nrow=med.dia, ncol=(length(x)/med.dia))
    for(i in 2:(length(x)/med.dia))
        {
        dia[,i]=x[(med.dia*(i-1)+1):((med.dia*(i-1))+med.dia)]
        dia.x[,i]=x[((med.dia*(i))+1):((med.dia*(i))+med.dia)]
        duplo[,i+1]=c(dia[,i], dia.x[,i])
        }
    matriz.d=matrix(c(1:(ncol(duplo))), (ncol(duplo)))
    layout(matriz.d, heights=rep(2, nrow(matriz.d)))
    par(mar=c(0,0,0,0), bty="n", lend="butt", oma=c(6,3,5,1), las=1,
cex=cex)
    if(lim.max=="n"|lim.min=="n")
        {
        plot(duplo[,1], type="h", lwd=5, xaxt="n", yaxt="n")
        for(i in 2:(ncol(duplo)-1))
            {
            plot(duplo[,i], type="h", lwd=5, xaxt="n", yaxt="n")
            axis(2, at=(c((max(duplo[,i])), (min(duplo[,i])))),  labels=(c("
", "")))
            axis(2, at=((max(duplo[,i]) - min(duplo[,i]))/2 +
min(duplo[,i])), labels=(i-1), lty="blank")
            }
        plot(duplo[,(ncol(duplo))], type="h", lwd=5, xaxt="n", yaxt="n")
        axis(2, at=(c((max(duplo[(1:med.dia),ncol(duplo)])),
(min(duplo[(1:med.dia),ncol(duplo)])))),  labels=(c(" ", "")))
        axis(2, at=((max(duplo[(1:med.dia),ncol(duplo)]) -
min(duplo[(1:med.dia),ncol(duplo)]))/2 + min(duplo[,i])),
labels=(ncol(duplo)-1), lty="blank")
        }
    else
        {
        plot(duplo[,1], type="h", lwd=5, xaxt="n", yaxt="n", ylim=c(lim.min,
lim.max))
        for(i in 2:(ncol(duplo)-1))
            {
            plot(duplo[,i], type="h", lwd=5, xaxt="n", yaxt="n",
ylim=c(lim.min, lim.max))
            axis(2, at=(c(lim.min, lim.max)),  labels=(c(" ", "")))
```

```
            axis(2, at=(((lim.max - lim.min)/2)+lim.min), labels=(i-1),
lty="blank")
            }
        plot(duplo[,(ncol(duplo))], type="h", lwd=5, xaxt="n", yaxt="n",
ylim=c(lim.min, lim.max))
        axis(2, at=(c((lim.min), lim.max)),  labels=(c(" ", "")))
        axis(2, at=(((lim.max - lim.min)/2)+lim.min),
labels=(ncol(duplo)-1), lty="blank")
        }
    axis(1, at=seq(0, med.dia*2, (60/intervalo*3)), outer=TRUE,
labels=c(seq(0, 24, 3), seq(3, 24, 3)), line=0.5, cex=0.8, xlab="Horas")
    title(main=titulo,  xlab="Horas", outer=TRUE)
    }
}
}
```

Arquivo do Exemplo

tuco.txt

Função do Plano B

Página de Ajuda

```
waveform               package:nenhum               R Documentation

Gráfico do tipo plexograma (waveform)

Description:

     Cria um gráfico do tipo plexograma (waveform) a partir de séries
temporais de vários dias, com intervalos regulares,
     com a possibilidade de plotar até duas variáveis no mesmo gráfico. Os
valores plotados no gráfico são a média dos
     valores de todos os dias para cada horário. O eixo x tem 24 horas e  o
y os valores da variável medida. A função
     também retorna um sumário do conjunto de dados (ver seção "Values" para
detalhes).

Usage:

     waveform(x, y=NULL, intervalo=5, linha="media", limmax.x=NULL,
limmin.x=NULL, limmax.y=NULL, limmin.y=NULL, ylab.x=NULL, ylab.y=NULL)

Arguments:
```

```
     x: Vetor numérico. Valores de uma variável medida em intervalos
regulares.
     y: Vetor numérico. Valores de uma variável medida em intervalos
regulares.
     intervalo: Numérico. Valor do intervalo entre medidas, em minutos.
     linha: Caracter. Se "media", plota uma linha horizontal correspondente
à média da variável. Se "mediana", plota uma linha
     correspondente à mediana. Se "n", nenhuma linha é plotada.
     limmax.x: Numérico. Limite máximo do eixo y, para o objeto "x".
     limmin.x: Numérico. Limite mínimo do eixo y, para o objeto "x".
     limmax.y: Numérico. Limite máximo do eixo y, para o objeto "y".
     limmin.y: Numérico. Limite mínimo do eixo y, para o objeto "y".
     ylab.x: Caracter. Legenda do eixo y para o objeto"x".
     ylab.y: Caracter. Legenda do eixo y para o objeto"y".

Details:

     É feita uma média, para cada horário, dos valores obtidos em todos os
dias e essas médias são plotadas contra um eixo x nos quais contam
     os horários. É possível plotar até duas variáveis no mesmo gráfico. Os
objetos devem ser vetores numéricos, correspondentes a medidas
     tomadas com intervalos regulares e iniciando à meia-noite.  O argumento
"linha" permite que seja desenhada uma linha com inclinação 0
     e intercepto correspondente à média ou mediana da variável.

Value:

     Para que a escala do eixo x corresponda aos horários reais, é
importante que as medidas que contam nos vetores iniciem à meia-noite.

     Um gráfico do tipo plexograma é criado no dispositivo gráfico.
     A função também retorna um sumário dos objetos, em formato de lista, na
qual:

  Média : A média de todos os valores do vetor.

  Mediana : Mediana de todos os valores do vetor.
  Máximo: Valor máximo dentre todos os valores do vetor.
  Mínimo: Valor mínimo dentre todos os valores do vetor.
  Média Mínima: A menor média dentre aquelas que aquelas que foram tomadas
para os valores de todos os dias, em cada horário.
  Fase da Média Mínima: Valor em minutos, correspondente ao horário da média
mínima.
  Média Máxima: A maior média dentre aquelas que aquelas que foram tomadas
para os valores de todos os dias, em cada horário.

  Fase da Média Mínima: Valor em minutos, correspondente ao horário da média
máxima.
  Amplitude: Diferença entre a média máxima e a média mínima.

Warning:
```

```
     Quando se deseja utilizar a função para duas variáveis, ambas devem ter
o mesmo intervalo entre medidas.

Author(s):

     Patricia Tachinardi Andrade Silva
     p.tachinardi@gmail.com

References:

     De Coursey PJ, Pius S, Sandlin C, Wethey D, Schull J. (1998).
Relationship of circadian temperature and activity rhythms in two rodent
species.
     Physiology & Behavior. 65(3):457–463.

See Also:

     plot(), para gráficos genéricos

Examples:

    ## Dados fictícios, obtidos em intervalos de 5 minutos por 15 dias
    exemplo= c(rep(c(runif(144, 20, 40), runif(144, 0, 10)), 15))
    waveform(exemplo, ylab.x="Exemplo")

    ## Dados ficitícios, obtidos em intervalos de 15 minutos por 30 dias
    objeto1= c(rep(c(runif(48, 30, 80), runif(48, 10, 50)), 30))
    objeto2= c(rep(c(runif(48, 70, 140), runif(48, 40, 60)), 30))
    waveform(objeto1, objeto2, ylab.x="Objeto 1", ylab.y="Objeto 2",
intervalo=15)
    # Agora com limites máximos e mínimos
    waveform(objeto1, objeto2, ylab.x="Objeto 1", ylab.y="Objeto 2",
intervalo=15, limmax.x=130, limmin.x=0, limmax.y=100, limmin.y=0 )
```

Código da Função

```
waveform= function(x, y=NULL, intervalo=5, linha="media", limmax.x=NULL,
limmin.x=NULL, limmax.y=NULL, limmin.y=NULL, ylab.x=NULL, ylab.y=NULL)
{
med.dia=1440/intervalo
x11()
resultados=matrix(nrow=288, ncol=(length(x)/med.dia)) # no lugar de 288,
colocar medidas.dia
resultados[,1]=x[1:med.dia] # tentar colocar isso no loop... ou não...
for(i in 1:(length(x)/med.dia-1))
    {
    resultados[,i+1]=x[((med.dia*i)+1):((med.dia*i)+med.dia)]
    }
media=apply(resultados, 1, mean)
```

```
# Sumario do objeto x
media.total=mean(resultados)
mediana.total= median(resultados)
max.total= max(resultados)
min.total=min(resultados)
max.media=max(media)
hor.max=which(media==max(media))*intervalo
hor.min=which(media==min(media))*intervalo
min.media=min(media)
amplitude= (max.media-min.media)
sumario=list("Média"=media.total, "Mediana"=media.total, "Máximo"=max.total,
"Mínimo"=min.total, "Média Mínima"= min.media,"Fase da Média Mínima (em
minutos)"= hor.min,
"Média Máxima"=max.media, "Fase da Média Máxima (em minutos)"= hor.max,
"Amplitude"= amplitude)

# No caso de só ter objeto x
if(is.null(y))
    {
    plot(media, xaxt="n", type="l", xlim=c(0,med.dia), ylim=c(limmin.x,
limmax.x), xlab="", ylab=ylab.x, bty="n")
    axis(1, at=seq(0, med.dia, (60/intervalo*3)), labels=seq(0, 24, 3),
line=0.5, cex=0.8)
    if (linha=="media")
        {
        abline(h=mean(resultados))
        }
    if(linha=="mediana")
        {
        abline(h=median(resultados))
        }
    title(xlab="Horas")
    return(sumario)
    }
# Com objeto x e y
else
    {
    resultados2=matrix(nrow=med.dia, ncol=(length(y)/med.dia))
    resultados2[,1]=y[1:med.dia]
    for(i in 1:(length(y)/med.dia-1))
        {
        resultados2[,i+1]=y[((med.dia*i)+1):((med.dia*i)+med.dia)]
        }
    media2=apply(resultados2, 1, mean)
    # Sumário do objeto y
    media.total2=mean(resultados2)
    mediana.total2= median(resultados2)
    max.total2= max(resultados2)
    min.total2=min(resultados2)
    max.media2=max(media2)
    hor.max2=which(media2==max(media2))*intervalo
```

```
    hor.min2=which(media2==min(media2))*intervalo
    min.media2=min(media2)
    amplitude2= (max.media2-min.media2)
    sumario2=list("Média"=media.total2, "Mediana"=mediana.total2,
"Máximo"=max.total2, "Média Mínima"= min.media2, "Fase da Média Mínima (em
minutos)"= hor.min2,
    "Média Máxima"=max.media2, "Fase da Média Máxima (em minutos)"= hor.max2
, "Amplitude"= amplitude2)
    # Lista com os dois sumarios:
    legy= deparse(substitute(y))
    legx= deparse(substitute(x))
    retorno=list(sumario, sumario2)
    names(retorno)=c(legx, legy)
    # O Gráfico
    par(mar=c(5, 4, 4, 8) + 0.1)
    plot(media, xaxt="n", type="l", xlim=c(0,med.dia), ylim=c(limmin.x,
limmax.x ), xlab="", ylab= ylab.x, bty="n")
    if (linha=="media")
        {
        abline(h=mean(resultados))
        }
    if(linha=="mediana")
        {
        abline(h=median(resultados))
        }
    par(new=TRUE)
    plot(media2, xaxt="n", type="l", xlim=c(0,med.dia), ylim=c(limmin.y,
limmax.y),xlab="", ylab="", yaxt="n", bty="n", col="red")
    if (linha=="media")
        {
        abline(h=mean(resultados2), col="red")
        }
    if(linha=="mediana")
        {
        abline(h=median(resultados2), col="red")
        }
    axis(4)
    mtext(ylab.y, side=4, line=2.5)
    axis(1, at=seq(0, med.dia, (60/intervalo*3)), labels=seq(0, 24, 3),
line=0.5, cex=0.8)
    title(xlab="Horas")
    return(retorno)
    }
}
```