



Simone Ota

Mestranda em Psicobiologia, UNIFESP

Meus exercícios

exec

Trabalho Final

Proposta A:

Muitos programas utilizados para leitura de registro de sono retornam uma tabela na qual cada coluna corresponde a um animal e cada linha corresponde a uma época (observação de alguns segundos) classificada como 1,2,3,4,5 ou 6.

A função a ser criada deve:

1. Substituir os números pelas siglas correspondentes na classificação (VIG(AWK) - vigília, V.Q.(QA)- vigília quieta, SOL (SWS)- sono de ondas lentas, SP (PS)- sono paradoxal, NC (UH)- não-classificado, ST (TS)- sono de transição)
2. Dividir o registro para que seja analisado de x em x horas (a ser definido pelo usuário)
3. Retornar em uma tabela o total de classificações de cada estado (VIG, SOL, etc), o tempo total em cada estado e a duração média de cada estado em cada x hora
4. A função ainda deverá retornar gráficos com as médias e erro-padrão dos resultados apresentados na tabela. Caso o usuário separe os animais em grupos, essas funções serão geradas para cada grupo.

Proposta B:

Muitos experimentos realizados no departamento envolvem a observação da interação de vários fatores no comportamento.

A função a ser criada deve:

1. Retornar o resultado da MANOVA (uma das análises mais utilizadas).
2. Realizar um teste post-hoc quando necessário.
3. Plotar um gráfico com as médias e erro-padrão de cada grupo.

Comentário Leandro

Simone, achei muito boa a ideia da função. Só precisa ficar claro como será essa separação dos animais em grupos. Será um novo atributo nos dados de entrada? Uma sugestão é focar primeiro na proposta mais simples e quando ela estiver pronta, tentar inserir a opção de agrupar os animais.

A função para o cálculo de MANOVA já existe no R. Minha sugestão é que você tente executar a primeira função, pois será algo que poderá ser usado por outras pessoas e acredito que não exista uma função pronta.

Comentários da função - Proposta A

Ainda faltam alguns ajustes, uma das dificuldades foi trabalhar com uma tabela de dados muito grande e tive que criar muitos "loops". Pretendo continuar trabalhando numa função que realize as tarefas propostas, talvez fazendo as análises separadas por animal e depois juntando tudo. Por enquanto a função retorna resultados totais (somando os dados de todos indivíduos) Também não consegui fazer com que a função leia a tabela de dados direto, o usuário deverá criar um objeto antes.

Página de Ajuda - Proposta A

sleeping.rat

package:nenhum

R Documentation

Parâmetros do sono

Description:

Produz uma tabela com a porcentagem do tempo de sono em cada estágio , o número de episódios de sono por estágio, o tempo médio (min) de sono em cada episódio de cada estágio, todos em um intervalo de duas horas. Também produz um gráfico de barras com os mesmos dados mostrados pela tabela.

Usage:

```
sleeping.rat(x)
```

Arguments:

x é um objeto (dataframe) criado previamente

Details:

Value:

São abertos 3 dispositivos com os gráficos referentes à porcentagem do tempo de sono, o número de episódios de sono por estágio, o tempo médio (min) de sono em cada episódio de cada estágio.

Retorna uma tabela na área de trabalho.

Warning:

Ainda faltam alguns ajustes... =(
 Por enquanto use dados com as mesmas dimensões do exemplo dado
 Author(s):

Simone Marie Ota (simone.ota@gmail.com)

References:

Timo-Iaria C, Negrão N, Schimidek WR, Hoshino K, Menezes CEL, Rocha TL.
 Phases and states of sleep in the rat. *Physiol Behav.* 1970; 5:1057-62.

Examples:

```
x=read.table("Sono ratos.txt",header=TRUE,sep="\t") ### Coloque o banco
de dados no formato .txt no lugar de 'Sono ratos'
sleeping.rat(x) ### Chame a função assim
```

Código da Função- Proposta A

```
#####
#                               #
#           Sleeping Rat       #
#                               #
#####

x=read.table("Sono ratos.txt",header=TRUE,sep="\t") ### Coloque o banco de
dados

sleeping.rat(x)### Chame a função assim

sleeping.rat= function(x)
{

a = as.matrix(x)           ###para substituir os valores
a[a == 1]= "AWK"
a[a == 2]= "SWS"
a[a == 3]= "PS"
a[a == 4]= "QA"
a[a == 5]= "UH"
a[a == 6]= "TS"

#10s= 1 classificação
#1min=60s = 6 clas
#1h=60min =3600s =360 clas
#0h~2h=1:720
#2h~4h=721:1440
#4h~6h=1441:2160
```

```
#6h~8h=2161:2880 ###esse registro para por ai
#8h~10h=2881:3600
#10h~12h=3601:4320

#vamos dividir em pedaços de 2h

a1=a[1:720,]
a2=a[721:1440,]
a3=a[1441:2160,]
a4=a[2161:2880,]

### Duplicando a matriz

b=a

b1=b[1:720,]
b2=b[721:1440,]
b3=b[1441:2160,]
b4=b[2161:2880,]

##### Substituindo valores consecutivos por 0 (NA não é boa opção)
##### Fazendo o ciclo para cada coluna nas primeiras 2h
for(n in 1:719) #A
{
  if(a1[n+1]==a1[n])
  {
    b1[n+1]=0
  }
  else
  {
    b1[n+1]=a1[n+1]
  }
}

for(n in 721:1439) #B
{
  if(a1[n+1]==a1[n])
  {
    b1[n+1]=0
  }
  else
  {
    b1[n+1]=a1[n+1]
  }
}

for(n in 1441:2159) #C
{
  if(a1[n+1]==a1[n])
  {
```

```
        b1[n+1]=0
      }
      else
      {
        b1[n+1]=a1[n+1]
      }
    }

for(n in 2161:2879) #D
{
  if(a1[n+1]==a1[n])
  {
    b1[n+1]=0
  }
  else
  {
    b1[n+1]=a1[n+1]
  }
}

for(n in 2881:3599) #E
{
  if(a1[n+1]==a1[n])
  {
    b1[n+1]=0
  }
  else
  {
    b1[n+1]=a1[n+1]
  }
}

for(n in 3601:4319) #F
{
  if(a1[n+1]==a1[n])
  {
    b1[n+1]=0
  }
  else
  {
    b1[n+1]=a1[n+1]
  }
}

for(n in 4321:5039) #G
{
  if(a1[n+1]==a1[n])
  {
    b1[n+1]=0
  }
  else
```

```
        {
        b1[n+1]=a1[n+1]
        }

    }

for(n in 5041:5759) #H
{
    if(a1[n+1]==a1[n])
    {
        b1[n+1]=0
    }
    else
    {
        b1[n+1]=a1[n+1]
    }
}

for(n in 5761:6479) #I
{
    if(a1[n+1]==a1[n])
    {
        b1[n+1]=0
    }
    else
    {
        b1[n+1]=a1[n+1]
    }
}

for(n in 6481:7199) #J
{
    if(a1[n+1]==a1[n])
    {
        b1[n+1]=0
    }
    else
    {
        b1[n+1]=a1[n+1]
    }
}

for(n in 7201:7919) #K
{
    if(a1[n+1]==a1[n])
    {
        b1[n+1]=0
    }
    else
    {
        b1[n+1]=a1[n+1]
    }
}
```

```
        }  
    }  
  
for(n in 7921:8639) #L  
    {  
        if(a1[n+1]==a1[n])  
        {  
            b1[n+1]=0  
        }  
        else  
        {  
            b1[n+1]=a1[n+1]  
        }  
    }  
  
for(n in 8641:9359) #M  
    {  
        if(a1[n+1]==a1[n])  
        {  
            b1[n+1]=0  
        }  
        else  
        {  
            b1[n+1]=a1[n+1]  
        }  
    }  
  
for(n in 9361:10079) #N  
    {  
        if(a1[n+1]==a1[n])  
        {  
            b1[n+1]=0  
        }  
        else  
        {  
            b1[n+1]=a1[n+1]  
        }  
    }  
  
##### Fazendo o ciclo para cada coluna das 2h~4h  
  
for(n in 1:719) #A  
    {  
        if(a2[n+1]==a2[n])  
        {  
            b2[n+1]=0  
        }  
        else  
        {  
            b2[n+1]=a2[n+1]  
        }  
    }
```

```
    }

for(n in 721:1439) #B
{
  if(a2[n+1]==a2[n])
  {
    b2[n+1]=0
  }
  else
  {
    b2[n+1]=a2[n+1]
  }
}

for(n in 1441:2159) #C
{
  if(a2[n+1]==a2[n])
  {
    b2[n+1]=0
  }
  else
  {
    b2[n+1]=a2[n+1]
  }
}

for(n in 2161:2879) #D
{
  if(a2[n+1]==a2[n])
  {
    b2[n+1]=0
  }
  else
  {
    b2[n+1]=a2[n+1]
  }
}

for(n in 2881:3599) #E
{
  if(a2[n+1]==a2[n])
  {
    b2[n+1]=0
  }
  else
  {
    b2[n+1]=a2[n+1]
  }
}
```



```
for(n in 3601:4319) #F
{
  if(a2[n+1]==a2[n])
  {
    b2[n+1]=0
  }
  else
  {
    b2[n+1]=a2[n+1]
  }
}

for(n in 4321:5039) #G
{
  if(a2[n+1]==a2[n])
  {
    b2[n+1]=0
  }
  else
  {
    b2[n+1]=a2[n+1]
  }
}

for(n in 5041:5759) #H
{
  if(a2[n+1]==a2[n])
  {
    b2[n+1]=0
  }
  else
  {
    b2[n+1]=a2[n+1]
  }
}

for(n in 5761:6479) #I
{
  if(a2[n+1]==a2[n])
  {
    b2[n+1]=0
  }
  else
  {
    b2[n+1]=a2[n+1]
  }
}

for(n in 6481:7199) #J
{
  if(a2[n+1]==a2[n])
```

```
        {
        b2[n+1]=0
        }
        else
        {
        b2[n+1]=a2[n+1]
        }
    }

for(n in 7201:7919) #K
{
    if(a2[n+1]==a2[n])
    {
        b2[n+1]=0
    }
    else
    {
        b2[n+1]=a2[n+1]
    }
}

for(n in 7921:8639) #L
{
    if(a2[n+1]==a2[n])
    {
        b2[n+1]=0
    }
    else
    {
        b2[n+1]=a2[n+1]
    }
}

for(n in 8641:9359) #M
{
    if(a2[n+1]==a2[n])
    {
        b2[n+1]=0
    }
    else
    {
        b2[n+1]=a2[n+1]
    }
}

for(n in 9361:10079) #N
{
    if(a2[n+1]==a2[n])
    {
        b2[n+1]=0
```

```
        }
        else
        {
            b2[n+1]=a2[n+1]
        }
    }

#####Fazendo o ciclo para cada coluna das 4h~6h

for(n in 1:719) #A
{
    if(a3[n+1]==a3[n])
    {
        b3[n+1]=0
    }
    else
    {
        b3[n+1]=a3[n+1]
    }
}

for(n in 721:1439) #B
{
    if(a3[n+1]==a3[n])
    {
        b3[n+1]=0
    }
    else
    {
        b3[n+1]=a3[n+1]
    }
}

for(n in 1441:2159) #C
{
    if(a3[n+1]==a3[n])
    {
        b3[n+1]=0
    }
    else
    {
        b3[n+1]=a3[n+1]
    }
}

for(n in 2161:2879) #D
{
    if(a3[n+1]==a3[n])
    {
        b3[n+1]=0
    }
}
```

```
        else
        {
        b3[n+1]=a3[n+1]
        }
    }

for(n in 2881:3599) #E
{
    if(a3[n+1]==a3[n])
    {
        b3[n+1]=0
    }
    else
    {
        b3[n+1]=a3[n+1]
    }
}

for(n in 3601:4319) #F
{
    if(a3[n+1]==a3[n])
    {
        b3[n+1]=0
    }
    else
    {
        b3[n+1]=a3[n+1]
    }
}

for(n in 4321:5039) #G
{
    if(a3[n+1]==a3[n])
    {
        b3[n+1]=0
    }
    else
    {
        b3[n+1]=a3[n+1]
    }
}

for(n in 5041:5759) #H
{
    if(a3[n+1]==a3[n])
    {
        b3[n+1]=0
    }
    else
    {
```

```
                b3[n+1]=a3[n+1]
            }
        }

for(n in 5761:6479) #I
{
    if(a3[n+1]==a3[n])
    {
        b3[n+1]=0
    }
    else
    {
        b3[n+1]=a3[n+1]
    }
}

for(n in 6481:7199) #J
{
    if(a3[n+1]==a3[n])
    {
        b3[n+1]=0
    }
    else
    {
        b3[n+1]=a3[n+1]
    }
}

for(n in 7201:7919) #K
{
    if(a3[n+1]==a3[n])
    {
        b3[n+1]=0
    }
    else
    {
        b3[n+1]=a3[n+1]
    }
}

for(n in 7921:8639) #L
{
    if(a3[n+1]==a3[n])
    {
        b3[n+1]=0
    }
    else
    {
        b3[n+1]=a3[n+1]
    }
}
```

```
for(n in 8641:9359) #M
{
    if(a3[n+1]==a3[n])
    {
        b3[n+1]=0
    }
    else
    {
        b3[n+1]=a3[n+1]
    }
}

for(n in 9361:10079) #N
{
    if(a3[n+1]==a3[n])
    {
        b3[n+1]=0
    }
    else
    {
        b3[n+1]=a3[n+1]
    }
}

#####Fazendo o ciclo para cada coluna das 6h~8h

for(n in 1:719) #A
{
    if(a4[n+1]==a4[n])
    {
        b4[n+1]=0
    }
    else
    {
        b4[n+1]=a4[n+1]
    }
}

for(n in 721:1439) #B
{
    if(a4[n+1]==a4[n])
    {
        b4[n+1]=0
    }
    else
    {
        b4[n+1]=a4[n+1]
    }
}
```

```
for(n in 1441:2159) #C
{
  if(a4[n+1]==a4[n])
  {
    b4[n+1]=0
  }
  else
  {
    b4[n+1]=a4[n+1]
  }
}

for(n in 2161:2879) #D
{
  if(a4[n+1]==a4[n])
  {
    b4[n+1]=0
  }
  else
  {
    b4[n+1]=a4[n+1]
  }
}

for(n in 2881:3599) #E
{
  if(a4[n+1]==a4[n])
  {
    b4[n+1]=0
  }
  else
  {
    b4[n+1]=a4[n+1]
  }
}

for(n in 3601:4319) #F
{
  if(a4[n+1]==a4[n])
  {
    b4[n+1]=0
  }
  else
  {
    b4[n+1]=a4[n+1]
  }
}

for(n in 4321:5039) #G
{
  if(a4[n+1]==a4[n])
```

```
        {
            b4[n+1]=0
        }
        else
        {
            b4[n+1]=a4[n+1]
        }
    }

for(n in 5041:5759) #H
{
    if(a4[n+1]==a4[n])
    {
        b4[n+1]=0
    }
    else
    {
        b4[n+1]=a4[n+1]
    }
}

for(n in 5761:6479) #I
{
    if(a4[n+1]==a4[n])
    {
        b4[n+1]=0
    }
    else
    {
        b4[n+1]=a4[n+1]
    }
}

for(n in 6481:7199) #J
{
    if(a4[n+1]==a4[n])
    {
        b4[n+1]=0
    }
    else
    {
        b4[n+1]=a4[n+1]
    }
}

for(n in 7201:7919) #K
{
    if(a4[n+1]==a4[n])
    {
        b4[n+1]=0
    }
}
```



```
        }
        else
        {
            b4[n+1]=a4[n+1]
        }
    }

for(n in 7921:8639) #L
{
    if(a4[n+1]==a4[n])
    {
        b4[n+1]=0
    }
    else
    {
        b4[n+1]=a4[n+1]
    }
}

for(n in 8641:9359) #M
{
    if(a4[n+1]==a4[n])
    {
        b4[n+1]=0
    }
    else
    {
        b4[n+1]=a4[n+1]
    }
}

for(n in 9361:10079) #N
{
    if(a4[n+1]==a4[n])
    {
        b4[n+1]=0
    }
    else
    {
        b4[n+1]=a4[n+1]
    }
}

##### Agora contando a porcentagem, o tempo total e episódios

awk1=length(a1[a1=="AWK"])/length(a1) ### %awake
awk2=length(a2[a2=="AWK"])/length(a2)
awk3=length(a3[a3=="AWK"])/length(a3)
awk4=length(a4[a4=="AWK"])/length(a4)
```

```
sws1=length(a1[a1=="SWS"])/length(a1) ### %slow wave sleep
sws2=length(a2[a2=="SWS"])/length(a2)
sws3=length(a3[a3=="SWS"])/length(a3)
sws4=length(a4[a4=="SWS"])/length(a4)

ps1=length(a1[a1=="PS"])/length(a1) ### %paradoxical sleep
ps2=length(a2[a2=="PS"])/length(a2)
ps3=length(a3[a3=="PS"])/length(a3)
ps4=length(a4[a4=="PS"])/length(a4)

qa1=length(a1[a1=="QA"])/length(a1) ### %quiet awake
qa2=length(a2[a2=="QA"])/length(a2)
qa3=length(a3[a3=="QA"])/length(a3)
qa4=length(a4[a4=="QA"])/length(a4)

uh1=length(a1[a1=="UH"])/length(a1) ### %unhooked
uh2=length(a2[a2=="UH"])/length(a2)
uh3=length(a3[a3=="UH"])/length(a3)
uh4=length(a4[a4=="UH"])/length(a4)

ts1=length(a1[a1=="TS"])/length(a1) ### %transition sleep
ts2=length(a2[a2=="TS"])/length(a2)
ts3=length(a3[a3=="TS"])/length(a3)
ts4=length(a4[a4=="TS"])/length(a4)

awka1=length(a1[a1=="AWK"])/6 ##### tempo de "acordado" nas primeiras 2h em minutos
swsa1=length(b1[a1=="SWS"])/6
psa1=length(a1[a1=="PS"])/6
qaa1=length(a1[a1=="QA"])/6
uha1=length(a1[a1=="UH"])/6
tsa1=length(a1[a1=="TS"])/6

awka2=length(a2[a2=="AWK"])/6 ##### tempo de "acordado" das 2h~4h em minutos
swsa2=length(a2[a2=="SWS"])/6
psa2=length(a2[a2=="PS"])/6
qaa2=length(a2[a2=="QA"])/6
uha2=length(a2[a2=="UH"])/6
tsa2=length(a2[a2=="TS"])/6

awka3=length(a3[a3=="AWK"])/6 ##### tempo de "acordado" das 4h~6h em minutos
swsa3=length(a3[a3=="SWS"])/6
psa3=length(a3[a3=="PS"])/6
qaa3=length(a3[a3=="QA"])/6
uha3=length(a3[a3=="UH"])/6
tsa3=length(a3[a3=="TS"])/6

awka4=length(a4[a4=="AWK"])/6 ##### tempo de "acordado" das 6h~8h em
```

```
minutos
swsa4=length(a4[a4=="SWS"])/6
psa4=length(a4[a4=="PS"])/6
qaa4=length(a4[a4=="QA"])/6
uha4=length(a4[a4=="UH"])/6
tsa4=length(a4[a4=="TS"])/6

awkb1=length(b1[b1=="AwK"]) ##### episódios de "acordado" nas primeiras 2h
swsb1=length(b1[b1=="SWS"])
psb1=length(b1[b1=="PS"])
qab1=length(b1[b1=="QA"])
uhb1=length(b1[b1=="UH"])
tsb1=length(b1[b1=="TS"])

awkb2=length(b2[b2=="AwK"]) ##### episódios de "acordado" das 2h~4h
swsb2=length(b2[b2=="SWS"])
psb2=length(b2[b2=="PS"])
qab2=length(b2[b2=="QA"])
uhb2=length(b2[b2=="UH"])
tsb2=length(b2[b2=="TS"])

awkb3=length(b3[b3=="AwK"]) ##### episódios de "acordado" das 4h~6h
swsb3=length(b3[b3=="SWS"])
psb3=length(b3[b3=="PS"])
qab3=length(b3[b3=="QA"])
uhb3=length(b3[b3=="UH"])
tsb3=length(b3[b3=="TS"])

awkb4=length(b4[b4=="AwK"]) ##### episódios de "acordado" das 6h~8h
swsb4=length(b4[b4=="SWS"])
psb4=length(b4[b4=="PS"])
qab4=length(b4[b4=="QA"])
uhb4=length(b4[b4=="UH"])
tsb4=length(b4[b4=="TS"])

##### Calculando o tempo médio por episódio a cada 2h

t.awk1=awka1/awkb1
t.awk2=awka2/awkb2
t.awk3=awka3/awkb3
t.awk4=awka4/awkb4

t.sws1=swsa1/swsb1
t.sws2=swsa2/swsb2
t.sws3=swsa3/swsb3
t.sws4=swsa4/swsb4

t.ps1=psa1/psb1
t.ps2=psa2/psb2
t.ps3=psa3/psb3
t.ps4=psa4/psb4
```

```
t.qa1=qaa1/qab1
t.qa2=qaa2/qab2
t.qa3=qaa3/qab3
t.qa4=qaa4/qab4

t.uh1=uha1/uhb1
t.uh2=uha2/uhb2
t.uh3=uha3/uhb3
t.uh4=uha4/uhb4

t.ts1=tsa1/tsb1
t.ts2=tsa2/tsb2
t.ts3=tsa3/tsb3
t.ts4=tsa4/tsb4

##### Criando alguns objetos para facilitar

Awake=c(awk1,awk2,awk3,awk4)
Slow.wave.sleep=c(sws1,sws2,sws3,sws4)
Paradoxical.sleep=c(ps1,ps2,ps3,ps4)
Quiet.awake=c(qa1,qa2,qa3,qa4)    ###No exemplo retorna 0, essa classificação
não foi utilizada
Unhooked=c(uh1,uh2,uh3,uh4)
Transition.sleep=c(ts1,ts2,ts3,ts4)

Awake.tempomedio= c(t.awk1,t.awk2,t.awk3,t.awk4)
Slow.wave.sleep.tempomedio= c(t.sws1,t.sws2,t.sws3,t.sws4)
Paradoxical.sleep.tempomedio= c(t.ps1,t.ps2,t.ps3,t.ps4)
Quiet.awake.tempomedio= c(t.qa1,t.qa2,t.qa3,t.qa4)
Unhooked.tempomedio= c(t.uh1,t.uh2,t.uh3,t.uh4)
Transition.sleep.tempomedio= c(t.ts1,t.ts2,t.ts3,t.ts4)

Awake.episode= c(awkb1,awkb2,awkb3,awkb4)
Slow.wave.sleep.episode= c(swsb1,swsb2,swsb3,swsb4)
Paradoxical.sleep.episode= c(psb1,psb2,psb3,psb4)
Quiet.awake.episode= c(qab1,qab2,qab3,qab4)
Unhooked.episode= c(uhb1,uhb2,uhb3,uhb4)
Transition.sleep.episode= c(tsb1,tsb2,tsb3,tsb4)

### Graficos

grafico.sono =x11()
par(mfrow=c(2,3))
barplot(Awake, main="Awake",xlab="",ylab="%Awake", ylim= c(0,1))
axis(1,1:4, c("2h", "4h", "6h", "8h" )[1:4])

barplot(Slow.wave.sleep, main="Slow wave sleep",xlab="", ylab="%Slow wave
sleep", ylim= c(0,1))
axis(1,1:4, c("2h", "4h", "6h", "8h" )[1:4])
```

```
barplot(Paradoxical.sleep, main="Paradoxical sleep",xlab="",
ylab="%Paradoxical sleep", ylim= c(0,1))
axis(1,1:4, c("2h", "4h", "6h", "8h" )[1:4])

barplot(Quiet.awake, main="Quiet awake",xlab="", ylab="%Quiet awake", ylim=
c(0,1))### Nos dados de exemplo não foi usada essa classificação
axis(1,1:4, c("2h", "4h", "6h", "8h" )[1:4])

barplot(Unhooked, main="Unhooked",xlab="", ylab="%Unhooked", ylim= c(0,1))
axis(1,1:4, c("2h", "4h", "6h", "8h" )[1:4])

barplot(Transition.sleep, main="Transition sleep",xlab="",ylab="%Transition
sleep", ylim= c(0,1))
axis(1,1:4, c("2h", "4h", "6h", "8h" )[1:4])

###
grafico.tempo.sono =x11()
par(mfrow=c(2,3))
barplot(Awake.tempomedio, main="Awake",xlab="",ylab=" Time Awake per
episode", ylim= c(0,5))
axis(1,1:4, c("2h", "4h", "6h", "8h" )[1:4])

barplot(Slow.wave.sleep.tempomedio, main="Slow wave sleep",xlab="",
ylab="Time in Slow wave sleep per episode", ylim= c(0,5))
axis(1,1:4, c("2h", "4h", "6h", "8h" )[1:4])

barplot(Paradoxical.sleep.tempomedio, main="Paradoxical sleep",xlab="",
ylab="Time in Paradoxical sleep per episode", ylim= c(0,5))
axis(1,1:4, c("2h", "4h", "6h", "8h" )[1:4])

barplot(Quiet.awake.tempomedio, main="Quiet awake",xlab="", ylab="Time in
Quiet awake per episode", ylim= c(0,5))### Nos dados de exemplo não foi
usada essa classificação
axis(1,1:4, c("2h", "4h", "6h", "8h" )[1:4])

barplot(Unhooked.tempomedio, main="Unhooked",xlab="", ylab="Time in Unhooked
per episode", ylim= c(0,5))
axis(1,1:4, c("2h", "4h", "6h", "8h" )[1:4])

barplot(Transition.sleep.tempomedio, main="Transition
sleep",xlab="",ylab="Time in Transition sleep per episode", ylim= c(0,5))
axis(1,1:4, c("2h", "4h", "6h", "8h" )[1:4])

###
grafico.epi.sono =x11()
par(mfrow=c(2,3))
barplot(c(awkb1,awkb2,awkb3,awkb4), main="Awake",xlab="",ylab="Number of
Awake", ylim= c(0,1000))
axis(1,1:4, c("2h", "4h", "6h", "8h" )[1:4])
```

```
barplot(c(swsb1,swsb2,swsb3,swsb4), main="Slow wave sleep",xlab="",
ylab="Number of Slow wave sleep", ylim= c(0,1000))
axis(1,1:4, c("2h", "4h", "6h", "8h" )[1:4])

barplot(c(psb1,psb2,psb3,psb4), main="Paradoxical sleep",xlab="",
ylab="Number of Paradoxical sleep", ylim= c(0,1000))
axis(1,1:4, c("2h", "4h", "6h", "8h" )[1:4])

barplot(c(qab1,qab2,qab3,qab4), main="Quiet awake",xlab="", ylab="Number of
Quiet awake", ylim= c(0,1000))### Nos dados de exemplo não foi usada essa
classificação
axis(1,1:4, c("2h", "4h", "6h", "8h" )[1:4])

barplot(c(uhb1,uhb2,uhb3,uhb4), main="Unhooked",xlab="", ylab="Number of
Unhooked", ylim= c(0,1000))
axis(1,1:4, c("2h", "4h", "6h", "8h" )[1:4])

barplot(c(tsb1,tsb2,tsb3,tsb4), main="Transition sleep",xlab="",ylab="Number
of Transition sleep", ylim= c(0,1000))
axis(1,1:4, c("2h", "4h", "6h", "8h" )[1:4])

### Tabela

Tabela=data.frame(Awake,Slow.wave.sleep,Paradoxical.sleep,Quiet.awake,
Unhooked,Transition.sleep,Awake.tempomedio,Slow.wave.sleep.tempomedio,
Paradoxical.sleep.tempomedio,Quiet.awake.tempomedio,Unhooked.tempomedio,
Transition.sleep.tempomedio,Awake.episode,Slow.wave.sleep.episode,
Paradoxical.sleep.episode,Quiet.awake.episode,Unhooked.episode,
Transition.sleep.episode,row.names=c("2h","4h","6h","8h"))
Tabela

##### Resultados
return(Tabela)

}
```

[sono_ratots.txt](#)

From:
<http://labbtrop.ib.usp.br/> - **Laboratório de Ecologia de Florestas Tropicais**

Permanent link:
http://labbtrop.ib.usp.br/doku.php?id=cursos:ecor:05_curso_antigo:r2011:alunos:trabalho_final:simone:start

Last update: **2020/07/27 18:48**