German Villanueva

Licenciado em biologia da Universidade Distrital Francisco Jose de Caldas na Colômbia. Mestrado em Biologia Animal na área de concentração em Biodiversidade Animal na Universidade Estadual de Campinas (UNICAMP). Atualmente Doutorando em Biologia Animal na UNICAMP.

Meu projeto de pesquisa no doutorado é sobre mecanismos de coexistência em comunidade de aranhas. Trabalhei com ecologia populacional, diversidade de aranhas orbiculares e com interações planta-aranha.

exec

Trabalho final

Proposta A Second Round

Proposta B

Help da função

```
pheno.correla                package: nenhum (ver details)         R
Documentation


  Analises fenológico de uma variável resposta numérica temporal e
correlação desta
  variável
  com variáveis preditoras temporais

Description:
  A função analisa se há sincronia entre uma variável resposta temporal e
outras
  variáveis explicativas temporais. Alem disso, a função realiza testes
circulares
  para saber se os dados registrados (variável resposta e preditoras) são
  homogéneos ao longo do ano ou se tem uma tendencia ou direção para um mês
no ano.
Usage:
  pheno.correla(x,rmNA=TRUE)
Arguments:

  x: Objeto de classe data-frame contento os dados brutos. As colunas devem
ter a
  mesma quantidade de observações (ver Details).
```

```
  rmNA: argumento logico para colocar se o data-frame tem ou não dados
faltantes nas
  colunas.

Details:

  O data-frame inserido (x) deve ter exatamente 6 colunas. A primeira coluna
deve ser
  de classe categórica contendo o ano ou os anos de estudo, a segunda coluna
sera
  categórica contendo os meses, a terceira coluna sera a variável resposta
temporal
  numérica, quarta quinta e sexta coluna serão variáveis numéricas temporais
(por
  exemplo, temperatura, precipitação, etc.) que serão correlacionadas com a
variável
  resposta.
  Todas as colunas deveram ter a mesma longitud de dados.
  O data-frame deve ter um número de linhas igual ou maior a 12. Ou seja,
deve existir
  no mínimo um ano de estudo sendo cada mês uma observação.
  Para que a função rode corretamente, é necesario que os meses introducidos
na coluna
  dois sejam as letras minusculas alfabéticas organizadas (ver "exemplos")
da
  seguinte forma:
  janeiro   =  a
  fevereiro =  b
  março     =  c
  abril     =  d
  .
  .
  .
  outubro   =  j
  novembro  =  k
  dezembro  =  l
  posteriormente a função mesma trocara esas letras pelos nomes dos meses
nos
  resultados e nos gráficos finais que o usuario/a obterá.
  Com o data-frame inserido corretamente, a função irá calcular a média das
variáveis
  resposta e preditoras por mês e o valor de correlação entre as variáveis
mediante o
  o coeficiente de correlação de Spearman.
  Posteriormente, serão calculados alguns parámetros circulares necesarios
para criar
  os gráficos e necesarios para observar possíveis padrões nos dados (ex.
Teste de
  Kuiper e de Rayleigh).
  Os gráficos e os testes circulares precisaram do pacote "circular". No
```

```
entanto, a
  função detetara se o usuario/a tem ou não o pacote. Se ele/ela não tem o
pacote,
  a função instalara o pacote "circular" para as analises pertinentes.
Value:

   comp1: Data-frame com as seguintes colunas: primeira coluna mostrando que
variável
   preditora esta sendo correlacionada com a variável resposta; segunda
coluna
   mostrando o valor do coeficiente de correlação entre as variáveis
(valores desde
   -1 até 1).
   comp2: lista com os seguintes valores: (1) Media dos valores numéricos de
todas as
   variareis (resposta e preditoras) por mês. (2) Valores dos testes
circulares de
   Rayleigh, Kuiper e comprimento do vetor medio (que indica a força da
direção
   dos dados) para as variáveis resposta e preditoras.
   comp3: Gráfico circular dos dados da variável resposta e preditoras
indicando
   os valores por mês e o vetor medio.
Author(s):

   German Antonio Villanueva Bonilla
References:

   -Hudson, I.L., & Keatley, M.R. (2009). Phenological research: methods for
   environmental and climate change analysis. Springer Science & Business
Media.
   -Sokal, R.R., Rohlf, F.J.. (1994). Biometry: the Principles and Practice
of
   Statistics in Biological Research. W. H. Freeman and Company, New York.
   -Zar, J.H. (1998). Biostatistical Analysis. Prentice Hall, Upper Saddle
River,
   New Jersey
   -
https://pt.wikipedia.org/wiki/Coeficiente_de_correla%C3%A7%C3%A3o_de_postos_
de_Spearman.

   -https://pt.wikipedia.org/wiki/Teste_de_Kuiper
Examples:

##(1) Exemplo de um data-frame com valores aleatorios em todas as variáveis
## e com dados faltantes em algumas colunas (NAs).
   abundancia=c(30,34,33,33,45,60,56,68,89,88,130,130,runif(12,90,130))
   temperatura=c(5,20,20,23,25,27,30,35,48,49,40,41,runif(12,41,80))
   precipitacao=c(2,5,7,13,14,23,23,25,36,38,40,47,runif(12,47,81))
   presas=c(6,6,7,20,23,22,30,36,48,46,55,55,runif(12,55,80))
   ano=rep(c("2015","2016"),each=12)
```

```
   meses=rep(letters[1:12],2)
   meses
   data=data.frame(ano,meses,abundancia,temperatura,precipitacao,presas)
   str(data)
   data[1,3]=NA
   data[3,3]=NA
   data[4,5]=NA
   data
   pheno.correla(data,rmNA=TRUE)
############
#(2) Este exemplo de data-frame tem a coluna presas (variável preditora)
##com valores altos em fevereiro e janeiro para ver a mudança no
##gráfico circular
  abundancia=c(30,34,33,33,45,60,56,68,89,88,130,130,runif(12,90,130))
  temperatura=c(5,20,20,23,25,27,30,35,48,49,40,41,runif(12,41,80))
  precipitacao=c(2,5,7,13,14,23,23,25,36,38,40,47,runif(12,47,81))
  presas=c(6,6,7,20,23,22,30,36,48,46,55,55,runif(12,55,80))
  ano=rep(c("2015","2016"),each=12)
  meses=rep(letters[1:12],2)
presas1=c(58,200,10,3,2,10,11,10,9,10,40,1,55,160,2,5,7,13,14,23,23,10,40,1)
  data2=data.frame(ano,meses,abundancia,temperatura,precipitacao,presas1)
  data2
  pheno.correla(data2,rmNA=TRUE)
############
#(3)Este exemplo tem um ano e um mês de dados (13 observações). É para
mostrar
##que a função roda com um número de observações igual ou maior a 12
observações
##que representaria um ano ou mais de dados.
##Alem disso, os valores para as variáveis preditoras "temp" e "pre" tem uma
##distribuição normal para mostrar como fica os gráficos circulares.

  ano=rep(2011,each=13)
  mes=c(letters[1:12],"a")
  abu=runif(13,80,180)
  prec=runif(13,80,180)
  temp=rnorm(13,25,2)
  pre=rnorm(13,130,2)
  dinamic1=data.frame(ano,mes,abu,prec,temp,pre)
  dinamic1
  pheno.correla(dinamic1,rmNA=TRUE)
########
#(4)Exemplo com mais de dois anos de estudo

  ano2=rep(2011,each=30)
  mes2=c(rep(letters[1:12],2),letters[1:6])
  abu2=runif(30,80,180)
  prec2=runif(30,80,180)
  temp2=rnorm(30,25,2)
  pre2=rnorm(30,130,2)
```

```
  dina2=data.frame(ano2,mes2,abu2,prec2,temp2,pre2)
  dina2
  pheno.correla(dina2,rmNA=TRUE)
```

Código da função

```
pheno.correla=function (x,rmNA=TRUE)    #os argumentos da função aceitaram
um data-frame (x) e a remoção dos dados faltantes (NA)
    {
#Decidi colocar aqui uma mensagem que falei como deve ser introducidos os
dados para qeu rode corretamente e para que os testes e as respostas sejam
as desejadas.
warning("Para que a função rode de forma correta o data.frame deve ter seis
colunas. \nA primeira coluna deve ser classe categorica contendo o ano ou os
anos de estudo, \nA segunda coluna sera categorica contendo os meses
representadas por letras minusculas consecutivas sendo
janeiro(a),fevereiro(b),março(c).....novembro(k),dezembro(l) \nA terceira
coluna sera a variável resposta temporal numerica, \nA quarta quinta e sexta
coluna serão variáveis numéricas temporais (por exemplo, temperatura,
precipitação, etc.) que serão correlacionadas com a variável resposta. \n
Todas as colunas deveram ter a mesma longitud de dados.")
    if(rmNA==TRUE)             #Aqui estoy falando que se "rmNA" é
verdadeiro delete os dados faltantes (NA).
        {
        dados=(na.omit(x))        #Criando um objeto que omita do data-frame
os NA
        dim <- dim(x)-dim(dados)     #para saber o numero total de NA
excluidos do data-frame
            n.NA <- dim[1]           #numero de NA excluidos do data-frame
            cat("Valores NA excluídosn",n.NA,"n","n") #aqui vou colocar na
que apareça na área de trabalho a mensagem de quantos valores NA foram
excluidos
        }
    else                         #aqui se ya foram excluidos os NA ou se não
teve necesidade de excluir nada
        {
        dados=x                #o novo data-frame sem valores NA para fazer
todas as analises
            warning("Se o seu data.frame tiver valores faltantes, a função
não irá rodar. O argumento rmNA deve ser verdadeiro\n",
call.=FALSE,immediate.=TRUE)
            }
        ##para todas as correlações eu decidi fazer "spearman" como metodo
porqeu ele não precisa da premisa de correlação linear entre as variaveis
numericas relacionadas, e segundo a literatura, é mais robusto para as
analises.
        spear.co1=cor(dados[,3],dados[,4],method ="spearman")    #primeiro
teste de correlação entre a vaiavel resposta e a primeira variavel preditora
```

```
numerica.
        spear.co2=cor(dados[,3],dados[,5],method ="spearman")    #segundo
teste de correlação entre a vaiavel resposta e a segunda variavel preditora
numerica.
        spear.co3=cor(dados[,3],dados[,6],method ="spearman")    #terceiro
teste de correlação entre a vaiavel resposta e a terceira variavel preditora
numerica.
        variavel.correlacionada=c(names(dados[4]),
names(dados[5]),names(dados[6])) #criando o objeto com as varaiveis
correlacionadas
        valor.teste=c(spear.co1,spear.co2,spear.co3)
#criando o objeto om os valores dos testes
        corre.total=data.frame(variavel.correlacionada,valor.teste)
#data-frame com os valores finales e nomes dos testes de correlação
                #agora vou verificar na função se esta instalado o packote
"circular" que sera utilizado para realizar os graficos circulares das
variaveis resposta e preditoras
                #Criei a função 'instalados' para verificar se o pacote
'circular' já está instalado ou não. (esta parte da função foi baseada no
script da Leticia Zimback)
            instalados <- function(pacotes)
                {
                 is.element(pacotes, installed.packages()[,1])    #a função
'is.element' é para conseguir obter um vetor com o nome dos pacotes
instalados.
            }
            if(instalados("circular")==FALSE)    #aqui se não esta instalado
o pacote circular a função instalara
                {
                install.packages("circular")        #aqui instala o pacote
com a função "install.packages"
                }
                if(instalados("circular")==TRUE)    #se o pacote ja esta
instalado ele le o pacote
                {
                require("circular", quietly=TRUE)    #aqui ele esta lendo o
pacote circula ja instalado
                }
            #agora vou comezar tirar os valores necesarios para fazer os
calculos para o plot e testes de uniformidad circular
            mean.abund= round(tapply(dados[,3],dados[,2],mean),0)
#aqui eu crie um objeto com a media da abundancia por mes porqeu sera
necessario para plotar no grafico circular
            mean.abund1=as.data.frame.table(mean.abund)           #Aqui
transformei o vector anterior para um data-frame qeu sera apresentado como
resultado junto com os outros dados
mean.abund1$Var1=levels=c("jan","fev","mar","abr","mai","jun","jul","ago","s
et","out","nov","dez")    #aqui eu troquei as letras que representam os
meses por os nomes abreviados dos meses
            mean.var1= round(tapply(dados[,4],dados[,2],mean),0)
```

```
#aqui eu crie um objeto com a media da variavel 1 por mes porqeu sera
necessario para plotar no grafico circular
            mean.var1.1=as.data.frame.table(mean.var1)           #Aqui
transformei o vector anterior para um data-frame qeu sera apresentado como
resultado junto com os outros dados
mean.var1.1$Var1=levels=c("jan","fev","mar","abr","mai","jun","jul","ago","s
et","out","nov","dez")    #aqui eu troquei as letras que representam os
meses por os nomes abreviados dos meses

            mean.var2= round(tapply(dados[,5],dados[,2],mean),0)
#aqui eu crie um objeto com a media da variavel 2 por mes porqeu sera
necessario para plotar no grafico circular
            mean.var2.1=as.data.frame.table(mean.var2)           #Aqui
transformei o vector anterior para um data-frame qeu sera apresentado como
resultado junto com os outros dados
mean.var2.1$Var1=levels=c("jan","fev","mar","abr","mai","jun","jul","ago","s
et","out","nov","dez")    #aqui eu troquei as letras que representam os
meses por os nomes abreviados dos meses

            mean.var3= round(tapply(dados[,6],dados[,2],mean),0)
#aqui eu crie um objeto com a media da variavel 3 por mes porqeu sera
necessario para plotar no grafico circular
            mean.var3.1=as.data.frame.table(mean.var3)           #Aqui
transformei o vector anterior para um data-frame qeu sera apresentado como
resultado junto com os outros dados
mean.var3.1$Var1=levels=c("jan","fev","mar","abr","mai","jun","jul","ago","s
et","out","nov","dez")    #aqui eu troquei as letras que representam os
meses por os nomes abreviados dos meses

                #vou cria agora alguns vetores necesarios para o plot
circular
                angulo=seq(0,330,by=30)    #vetor de doze números espaçados
igualmente que serão posicionados radialmente na circunferência
                angulo.rad=rad(angulo)    #é depois o vetor anterior foi
transformado em radianes

abund.feno=circular(rep(angulo.rad,mean.abund),units=c("radians"),rotation="
counter")    #transformação do vetor que tem a media do valor numerico
resposta por mes em um vetor circular que possa ser ploteado
                media.abun=mean(abund.feno)    #Calculei a média para obter
o vetor que indica a direção das observações na variavel resposta.

var1.feno=circular(rep(angulo.rad,mean.var1),units=c("radians"),rotation="co
unter")        #transformação do vetor que tem a media do valor numerico
preditor 1 por mes em um vetor circular que possa ser ploteado
                media.var1=mean(var1.feno)    #Calculei a média para obter o
vetor que indica a direção das observações na variavel preditoria 1.



var2.feno=circular(rep(angulo.rad,mean.var2),units=c("radians"),rotation="co
unter")        #transformação do vetor que tem a media do valor numerico
```

```
preditor 2 por mes em um vetor circular que possa ser ploteado
                media.var2=mean(var2.feno)    #Calculei a média para obter o
vetor que indica a direção das observações na variavel preditoria 2.



var3.feno=circular(rep(angulo.rad,mean.var3),units=c("radians"),rotation="co
unter")        #transformação do vetor que tem a media do valor numerico
preditor 3 por mes em um vetor circular que possa ser ploteado
                media.var3=mean(var3.feno)    #Calculei a média para obter o
vetor que indica a direção das observações na variavel preditoria 3.



                meses <-
c("jan","fev","mar","abr","mai","jun","jul","ago","set","out","nov","dez")
#este vetor criado é para colocar como label no grafico crclar
                par(mfrow=c(2,2))        #aqui eu dividi a janela grafica em
4 porque são quatro graficos qeu serão feitos: (1)variavel
resposta;(2)variavel preditora 1;(3)variavel preditora 2;(4)variavel
preditora 3
                par(mar=c(1,1,1,1))    #para que os graficos tiveram espaço
suficiente eu coloquei margens ben estreitas
                ##nos graficos eu coloquei como nome principal o nome que
aparece na coluna da variavel resposta ou preditora segundo corresponda
                ##Para todos os diagramas de rosa, os argumentos são:
(1)conjunto de dados de classe circular; (2)"bins" é o numero de divisões
que tera o circulo; (3)"add" é para adicionar este diagrama em um grafico ja
existente; (4)"axes" aqui eu coloquie FALSE para qeu não coloque nada
                ##Para todos os graficos, a função arrows.circular tem como
argumentos: (1)os dados circulares como media; (2)"shrink" indica o
comprimento da linha; (3)a cor da linha, neste caso vermelho; (4)e
finalmente o comprimento das pontas da seta graficada.
                #plot (1) com a abundancia por mes (variavel resposta
numerica temporal)
                comprimento.vector.r1=rho.circular(abund.feno)    #o
comprimento do vetor médio que indica a direção das observações. Este valor
vai de 0 até 1, onde 1 indica grande acumulo ou tendencia dos dados para uma
direção certa. No grafico o vetor sera maior enquanto o valor seja maior
tambem.
                plot(abund.feno,axes=F,main="Fenologia")        #Plot
circular dos dados da variavel resposta ja transformados, neste grafico nos
outros qeu não coloque os eixos para depois eu trocar pelo vetor "meses"
criado anteriormente
axis.circular(at=circular(rad(angulo)),labels=meses,units=c("radians"))
#aqui coloquie no eixo do grafico circular os meses do vetor ants criado e
falei para colocar no formato de radianes para qeu coincida
                rose.diag(abund.feno,bins=12,add=T,axes=F)    #Plotei este
diagrama em rosa para visualizar dentro do grafico circular antes ploteado
os valores numericos da variavel resposta por cada mes (na verdade a média
dos valores por cada mes)
                arrows.circular(x=media.abun, shrink=comprimento.vector.r1,
```

```
col="red",length=0.1)    #Esta função plota um vetor na direção média dos
dados no grafico anterior, entre maior o comprimento da linha maior sera a
tendencia dos dados para um mes no ano

                #plot (2) com os dados da variavel numero 1 por mes
(variavel preditora numerica temporal que estaria na coluna 4)
                comprimento.vector.r2=rho.circular(var1.feno)    #o
comprimento do vetor médio que indica a direção das observações. Este valor
vai de 0 até 1, onde 1 indica grande acumulo ou tendencia dos dados para uma
direção certa
                plot(var1.feno,axes=F,main=names(dados[4]))    #Plot
circular dos dados da variavel preditora 1 ja transformados
axis.circular(at=circular(rad(angulo)),labels=meses,units=c("radians"))
#Igual que no outro grafico, aqui coloquei os meses como eixos
                rose.diag(var1.feno,bins=12,add=T,axes=F)        #igual que
no diagrama de rosa anterior, plotie os valores da primeira variavel
preditoria por mes no grafico circular.
                arrows.circular(x=media.var1, shrink=comprimento.vector.r2,
col="red",length=0.1)    #Como no anterior, aqui tambem estara la linha
(vetor medio) em vermelho qeu indicara a direção dos dados no grafico
circular.

                #plot (3) com os dados da variavel numero 2 por mes
(variavel preditora numerica temporal que estaria na coluna 5)
                comprimento.vector.r3=rho.circular(var2.feno)    #o
comprimento do vetor médio que indica a direção das observações. Este valor
vai de 0 até 1, onde 1 indica grande acumulo ou tendencia dos dados para uma
direção certa
                plot(var2.feno,axes=F,main=names(dados[5]))    #Plot
circular dos dados da variavel preditora 2 ja transformados
axis.circular(at=circular(rad(angulo)),labels=meses,units=c("radians"))
#Igual que no outro grafico, aqui coloquei os meses como eixos
                rose.diag(var2.feno,bins=12,add=T,axes=F)        #igual que
no diagrama de rosa anterior, plotie os valores da segunda variavel
preditoria por mes no grafico circular.
                arrows.circular(x=media.var2, shrink=comprimento.vector.r3,
col="red",length=0.1)    #Como no anterior, aqui tambem estara la linha
(vetor medio) em vermelho qeu indicara a direção dos dados no grafico
circular.

                #plot (4) com os dados da variavel numero 3 por mes
(variavel preditora numerica temporal que estaria na coluna 6)
                comprimento.vector.r4=rho.circular(var3.feno)    #o
comprimento do vetor médio que indica a direção das observações. Este valor
vai de 0 até 1, onde 1 indica grande acumulo ou tendencia dos dados para uma
direção certa
                plot(var3.feno,axes=F,main=names(dados[6]))    #Plot
circular dos dados da variavel preditora 3 ja transformados
axis.circular(at=circular(rad(angulo)),labels=meses,units=c("radians"))
#Igual que no outro grafico, aqui coloquei os meses como eixos
                rose.diag(var3.feno,bins=12,add=T,axes=F)        #igual que
```

```
no diagrama de rosa anterior, plotie os valores da terceira variavel
preditoria por mes no grafico circular.
                arrows.circular(x=media.var3, shrink=comprimento.vector.r4,
col="red",length=0.1)    #Como no anterior, aqui tambem estara la linha
(vetor medio) em vermelho qeu indicara a direção dos dados no grafico
circular.
                par(mfrow=c(1,1))    #Aqui coloquei de novo o painel do
grafico com uma coluna e uma linha só.

        #Agora vou fazer os testes de uniformidade
        ##o primeiro é saber se os dados tem uma distribuição normal
circular (analoga com a distribuição normal de campana de Gauss).
        #para isso eu testo essa normalidade com o teste de melhor ajuste
circular (análogo com o Teste de Kolmogorov-Smirnov), que seria o teste de
Kuiper.
        #Valores do teste acima do valor crítico, indicam a existência de um
padrão e menor abrangência fenológica.
        kuiper.abund=kuiper.test(abund.feno, alpha=0.05)    ##Aqui é testado
se as medias das observações são iguais, se não são rejeito a hipoteses nula
(uniformidade). Hipoteses alterna é que existe uma tendencia dos dados é não
uma homogeneidae.
        kuiper.var1=kuiper.test(var1.feno, alpha=0.05)    ##Aqui fazo o
mesmo para testar a uniformidade da primeira variavel preditora
        kuiper.var2=kuiper.test(var2.feno, alpha=0.05)    ##Aqui fazo o
mesmo para testar a uniformidade da segunda variavel preditora
        kuiper.var3=kuiper.test(var3.feno, alpha=0.05)    ##Aqui fazo o
mesmo para testar a uniformidade da terceira variavel preditora

        #Agora vou fazer o teste de Rayliegh para saber se existe uma
direção ou tendencia dos dados e se é significativa ou não.

        rayleigh.abun=rayleigh.test(abund.feno)    #teste de rayleigh para a
varivel resposta temporal, valores inferiores no p= 0.05 indicam qeu existe
uma tendencia dos dados hacia uma direção
        rayleigh.var1=rayleigh.test(var1.feno)    #teste de rayleigh para a
varivel preditoria 1
        rayleigh.var2=rayleigh.test(var2.feno)    #teste de rayleigh para a
varivel preditoria 2
        rayleigh.var3=rayleigh.test(var3.feno)    #teste de rayleigh para a
varivel preditoria 3

##Agora vou fazer uma lista com os resultados qeu eu acho importantes
(testes de correlação, circulares, etc para o usuario ver.
##posteriormente, coloquie os nomes dentro da lista dos resultados
list.mean=list(corre.total,mean.abund1,mean.var1.1,mean.var2.1,mean.var3.1,k
uiper.abund,kuiper.var1,kuiper.var2,kuiper.var3,rayleigh.abun,rayleigh.var1,
rayleigh.var2,rayleigh.var3,comprimento.vector.r1,comprimento.vector.r2,comp
rimento.vector.r3,comprimento.vector.r4)
names(list.mean)=c("Coeficiente de correlação de Spearman da variavel
resposta","abundancia media por mês", names(dados[4]),
```

```
names(dados[5]),names(dados[6]),"Teste Kuiper para variavel resposta","Teste Kuiper para variavel preditora 1","Teste Kuiper para variavel preditora 2","Teste Kuiper para variavel preditora 3","Teste Rayleigh para variavel resposta","Teste Rayleigh para variavel preditora 3","Teste Rayleigh para variavel preditora 3","Teste Rayleigh para variavel preditora 3","Comprimento do vetor medio da variavel resposta","Comprimento do vetor medio da variavel preditora 1","Comprimento do vetor medio da variavel preditora 2","Comprimento do vetor medio da variavel preditora 3")
##Para finalizar a função return mostra na área de trabalho a o vetor lista que eu anteriormente criei com os resultados relevantes da função.
return(list.mean)
    }
```

Arquivo da função

funcao_corr.r

From:
http://ecor.ib.usp.br/ - **ecoR**

Permanent link:
http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2016:alunos:trabalho_final:germanvillanueva9:start

Last update: **2020/08/12 06:04**