

Gabriel Costa



Msc. Gabriel Brito Costa / Doutorando PPGI- EA Esalq/CENA- USP / Laboratório de Clima e Biosfera - LCB / Departamento de Ciências Atmosféricas/ IAG/ Universidade de São Paulo

Proposta de Trabalho Final

Principal

Fazer uma função para calcular intervalos de percentis de qualquer variável (no meu caso, NEE) automaticamente, sem a necessidade de repetir a operação para um intervalo diferente de uma segunda variável, sendo assim no código apenas dizer o intervalo da variável 2 e o intervalo de percentil que eu quero os dados. Isto serve para remover extremos e plotar uma curva de dependência luminosa (NEE X PAR)

Plano B

Uma função para plotar gráficos do vetor vento (direção do vento com setas indicativas e velocidade em um mesmo plot)

Comentários

Caro Gabriel,

Faltou os elementos básicos para avaliar a viabilidade da suas propostas. Quais os dados de entrada, argumentos e a saída da função. Além disso é importante definir os formatos e classes dos objetos. A princípio, a proposta A parece muito simples e demanda apenas a aplicação de uma função básica do R. O plano B parece um desafio mais adequado, mas de qq forma não é possível avaliar a proposta sem um pouco mais de informação além da tarefa que a função irá fazer. Aguardamos os seus ajustes para fazer a avaliação de uma proposta final.

— [Alexandre Adalardo de Oliveira](#) 2012/04/04 14:48

Resposta dos Comentários

Prezado Alexandre,

Talvez eu não tenho conseguido expressar o que de fato queria fazer no plano 1, que me seria algo bastante útil assim como o 2.

Enfim, no 2 eu tenho dados de direção e velocidade do vento em um arquivo de médias de 30 minutos, e a proposta seria criar uma função que plota os vetores da direção (com setas) e de velocidade tanto na escala diária (1 único plot mostra os padrões de velocidade e direção do vento de

determinado dia, fugindo um pouco do padrão “rosa dos ventos” e inovando com os vetores da direção) quanto na escala que eu desejar (a função teria a mobilidade de, caso eu queira plotar um longo período de dados, tipo 1 mês ou mesmo 1 ano, também gerar um plot similar ao da escala diária).

Resposta da Resposta

O plano 1 me parece realmente muito básico para o trabalho final. Ainda não consigo avaliar o gráfico que a função pretende fazer. Do jeito que está colocado pode ser um desafio interessante ou um tanto básico, não dá para saber. Vc. tem algum exemplo que possa me mostrar?

Seria algo com o windguru ou o que tem no site <http://www.642weather.com/weather/>?

- Os vetores seriam em que escala de tempo? a média de 30 minutos?
- Quais os eixos do gráfico? velocidade ~ tempo?
- Como seria o ajuste do gráfico para diferentes escalas de observação (dia, mes, ano)?

— [Alexandre Adalardo de Oliveira](#) 2012/04/05 13:10

Resposta da Resposta da resposta

Bom Alexandre, vamos esquecer o plano 1, acho que o plano 2 ganhou força.

* Os vetores seriam na escala de 30 minutos. São as observações que tenho, a função teria por objetivo fazer um plot destes dados, só que não como rosa de ventos (estilo radar), e sim com setas indicativas para cada horário e suas respectiva direção (e, em um plot abaixo, outro gráfico com a velocidade)

* O plot de cima não teria eixo y. Seria o “vetor vento”, com direcionamento para os pontos cardeais (Se a setinha está totalmente a 90° perpendicular ao eixo X, Voltada para cima, significa que aquele vento veio da direção sul, de 180°, consegue visualizar?)

No caso da velocidade, sim, esta em função do tempo. No eixo x se tem o tempo cronológico (horário da observação) e o dia correspondente. Seria 1 plot com 2 gráficos, o de cima direção e o de baixo velocidade (ou vice-versa).

* Em relação á escala de tempo, na minha área usamos muito o “dia do ano”, sendo assim o ajuste seria que, quando eu pedisse um plot de 1 mês de dados, ele me traria um gráfico com as direções (indicadas por vetores de seta) e velocidade e abaixo o dia do ano indicativo, bem como as horas do dia.

Sim, aquele exemplo do site é mais ou menos parecido, talvez eu tenha pensado em setas maiores do que aquele triangulozinho da figura.

Página de Ajuda

Wind.plot

package:unknown

R Documentation

~~Função para fazer um plot dinâmico de padrão de vento com setas indicando a direção ~~

Description:

~~Com dados de 30 minutos de direção e velocidade do vento, faz um plot do padrão de vento com setas indicando de onde o vento vem~~

Usage:

~~function WindPlot~~

Details:

~~O script chama 2 funções de fora do mesmo, a saber: "T0.HHMM" para a conversão de data usando o posix e "opteixolabels" para selecionar posições no eixo do timevector~~

Warning:

~~O script foi rodado usando o R version 2.12.1 (2010-12-16). Não estou certo se ele funciona em versões anteriores e acho que rodará em versões atualizadas do programa.

Em caso de não funcionar, verificar qual versão do R está se usando e baixar a versão 2.12.1 que tenho certeza funcionará. Um outro detalhe importante é que o script funcionou rodando 1 ano de dados, completo.

Não estou certo se ele funciona com arquivos de mais de 1 ano (poderia haver conflito com os dias do ano).

Neste caso o ideal para se rodar mais de 1 ano de dados é separar cada ano em um arquivo diferente, e se não tem dados no arquivo, ele não roda COM O CÓDIGO DE ERRO (NA) nem com valores faltantes (célula vazia).

O ideal é, em horários com ausência de dados, colocar um código de erro diferente, mas que você sabe que é erro (0, por exemplo)~~

Author(s):

~~Gabriel Brito Costa et al~~

Código da função

```
# carregando pacotes necessarios
```

```
library(fields)
library(sfsmisc)
library(plotrix)
```

```
## diretório de scripts e dados
path <- getwd()

## carrega funcoes necessarias, elas devem estar no mesmo diretório do
script.
source(paste(path, "To_HHMM.r", sep="/"))
source(paste(path, "opteixolabels.r", sep="/"))

## O arquivo de dados está em subdiretório chamado "DADOS_MINUTOS", ou seja
em uma pasta separada dentro da pasta onde estão os scripts
"Windplot", "To_HHMM" e "opteixolabels"
subdir <- "DADOS_MINUTOS"
filename <- paste(subdir, list.files(subdir)[length(list.files(subdir))],
sep="/")
input <- paste(path, filename, sep="/")
dados <- read.table(input, head=T, na.strings="NA",
blank.lines.skip = TRUE, fill=T)

names(dados) <- c("ANO", "MES", "DIA", "HORA", "MN", "VEL", "DIR")

## Construção de data correta
## gerando time stamp da data de referencia
## especificar data, hora e minuto inicial do arquivo
dtInicial <- as.POSIXct("2010-1-1 00:00:00 BRT")
dtFinal <- as.POSIXct("2010-12-31 23:30:00 BRT")

seqD <- seq(dtInicial, dtFinal, by='min')

## time stamp dos dados originais
#1) deixa HORA com 4 posicoes (preenche com zeros)
# 1 passa 0001, ...
hora.char <- as.character(dados[, "HORA"])
hora.char <- paste(substr(To_HHMM(hora.char), 1, 2),
substr(To_HHMM(hora.char), 3, 4), "00", sep=":")

data.char <- paste(dados[, "ANO"], dados[, "MES"], To_DD(dados[, "DIA"]), sep="-")
timestamp_orig <- paste(data.char, hora.char, "BRT", sep=" ")
timestamp_orig <- as.POSIXct(timestamp_orig)

# construindo uma nova matriz de dados baseado no tamanho de seqD
dadosok <- data.frame(matrix(NA, ncol=ncol(dados), nrow=length(seqD)))
names(dadosok) <- names(dados)

#####
# gerando vetores de u e v
u <- (-1*dados[, "VEL"]*sin(dados[, "DIR"]*(pi/180)))
v <- (-1*dados[, "VEL"]*cos(dados[, "DIR"]*(pi/180)))

#####
```

```
# sub-período para plot
# selecionando o(s) dia(s) para plot

WindPlot <- function(u, v, timestamp, whatdate, ndays=1, comp=0.02, len=0.2)
{
  pos <- which(format(timestamp,"%j") %in% format(whatdate,"%j"))

  # primeira posicao
  posi <- min(pos)
  # ultima
  posf <- posi + round(ndays*48)

  # definindo variaveis para plot
  tempo <- timestamp[posi:posf]
  uu <- u[posi:posf]
  vv <- v[posi:posf]
  VV <- sqrt(uu*uu+vv*vv)
  x <- seq(along=uu)
  y <- rep(1, length(x))

  # remover NAS
  if(sum(is.na(uu))>0 | sum(is.na(vv))>0)
  {
    onde <- complete.cases(cbind(u,v))
    uu[!onde] <- 0.1
    vv[!onde] <- 0.1
  }

  ## plot
  # os 2 parametros abaixo devem ser ajustados caso o ndays seja alterado
  #comp <- 200; len <- 0.0002
  op <- par(mfcol=c(2,1), mar=c(2,4,3,2), oma=c(3,0.5,1,1), cex.axis=1.2,
  cex.lab=1.2)
  plot(x, y, type="n", yaxt="n", frame=T, xaxt="n", ylab="Wind Vector",
  xlab="",main = "Wind Pattern")
  posicoes <- axTicks(1, axp=round(c(0,max(axTicks(1)),12)))
  arrow.plot(x,y,uu,vv, arrow.ex= comp , length=len*comp*10^1.5, col="black",
  true.angle = T, xpd=T)
  points(x,y,pch=20,cex=0.25,col="red")
  axis(1, at=posicoes, labels=opteixo(tempo, posicoes, opcao=7))
  #axis(3, at=posicoes, labels=opteixo(tempo, posicoes, opcao=1) )
  for(k in seq(1,length(x),by=360)[c(F,T)]) abline(v=k, lty=2,col="yellow")
  box(lwd=2)
  plot(x, VV, type="l",col="gray", axes=F, xlab="",ylab="Speed (m/s)")
  #Aqui será escolhido o formato do eixo x. OPÇÃO 7 É = dia/mês/hora/minuto.
  Para obter o eixo x em outro formato, mudar o numero da opção aqui, que foi
  especificada em "opeixolabels"
  axis(1, at=posicoes, labels=opteixo(tempo,posicoes,opcao=7))
  axis(2, labels=T); eaxis(4,labels=F)
  for(k in seq(1,length(x),by=360)[c(F,T)]) abline(v=k, lty=2,col="yellow")
  box(lwd=2)
```

```
mtext(paste("from", format(range(tempo, na.rm=T), "%F %H:%M")[1],  
"to", format(range(tempo, na.rm=T), "%F %H:%M")[2]),  
cex=0.7, side=1, line=4, adj=1)  
  
par(op)  
}  
#Aqui será escolhido o dia para o plot do Padrão de vento  
diainicial <- as.POSIXct("2010-10-29 00:00:00 BRT")  
WindPlot(u=u, v=v, timestamp=timestamp_orig,  
          whatdate=diainicial, ndays=1, comp=5, len=0.00066)  
#Lembrando que ele dará o formato de tempo conforme a escolha da opção em  
"axis,posição", sendo o formato da posição descrito na função "opteixo" .  
  
#Para selecionar um período maior de dias, mudar o ndays (numero de dias  
desejaveis no plot). Neste caso estou plotando 3 dias seguidos.  
x11()  
WindPlot(u=u, v=v, timestamp=timestamp_orig,  
          whatdate=diainicial, ndays=3, comp=20, len=0.000005)  
  
#Fim.
```

Arquivo da função

[windplott.zip](#)

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

Permanent link:
http://labtrop.ib.usp.br/doku.php?id=cursos:ecor:05_curso_antigo:r2013:alunos:trabalho_final:gabrielbc:start

Last update: 2020/07/27 21:46

