

```
#roteiro pratica
#elaborado por Marcio Pupin Mello (www.dsr.inpe.br/~mello)
#data: 27/Set/2012

#####
#antes de comecar
#####

#criar uma pasta chamada "aulaR" no computador
#no meu caso, criei em "/Users/mello/desktop/aulaR"

#baixar para dentro da pasta "aulaR" o arquivo "CursoR.zip" do endereco
#www.dsr.inpe.be/~mello/CursoR

#descompactar o arquivo "CursoR.zip" usando a senha "user"
#apos descompactado, devem aparecer 3 subpastas: (i) tabular; (ii) imagem; e (iii) shape.

#ABRIR O R!!!

#configurar o diretorio de trabalho para ("/Users/mello/desktop/aulaR")
setwd("/Users/mello/desktop/aulaR") #atencao para a barra que nao eh invertida
getwd() #apenas verificando o diretorio de trabalho atual
dir() #exibe o conteudo do diretorio de trabalho atual

#####
#dados tabulares
#####

#####
#usando o arquivo ./tabular/reg_lin.xls

#abra o arquivo usando o Excel ou similar
#vamos ler os dados para o R usando o comando "scan()"

?scan #com o help, aprenda a utilizar o comando!
x<-scan() #copiar os dados de x da planilha e colar no R
y<-scan() #copiar os dados de y da planilha e colar no R

dados1<-data.frame(temp=x,dilat=y) #organizando os dados na estrutura de data.frame
rm(x,y) #ja podemos apagar x e y, evitando acumular "lixo"

dados1 #vizualiza o objeto "dados1" que acabamos de criar
plot(dados1) #vizualiza graficamente o objeto
plot(dados1,main="Dados 1",xlab="Temperatura em graus Celsius","Dilatação em micrometros")

#vamos ajustar uma regressao linear!!! para isso, use o comando "lm()"
#nosso modelo linear eh  $y = b_0 + b_1*x$ 

reg<-lm(dilat~temp,dados1) #ajustando o modelo
abline(reg,col="red") #exibindo a reta no grafico
summary(reg) #visualizando alguns resultados da regressao
```

```

#vc tb pode usar o comando "predict()" para estimar novos valores de y com base em um x
observado
#use ?predict para aprender como

#####
#usando o arquivo ./tabular/dadosRS.xls

#abra o arquivo usando o Excel ou similar
#use o salvar como "texto separado por tabulacao"
#o novo arquivo deve chamar-se "dadosRS.txt" e estar dentro da subpasta "tabular"
#vamos importar o arquivo usando o read.table

?read.table #com o help, aprenda a utilizar o comando!
dados2<-read.table("./tabular/dadosRS.txt",header=T,dec=".") #lendo os dados do arquivo
head(dados2) #exibe apenas um pedaco do objeto "dados2"

summary(dados2) #exibe estatisticas descritivas do objeto
is.factor(dados2$class) #verificar se as classes tematicas estao definidas no R como fator
dados2$class #exibindo a coluna "class" do objeto "dados2"
is.factor(dados2$b1) #e quanto a b1, por exemplo?

#vamos criar um scatterplot matrix... o próprio comando plot ja se encarrega disso!
plot(dados2) #observe que ele incluiu a class como atributo
plot(dados2[,1:6],main="TM/Landsat-5") #plotando apenas as bandas espectrais

#uma opcao para diferenciar as classes por cores eh usar o comando "pairs()"
pairs(dados2[,1:6],main="TM/Landsat-5",pch=21,bg=c("red","green","blue")[unclass(dados2$class
)]) #pintando

#####
#dados GeoTiff
#####

#####
#usando o arquivo ./imagem/B4.tif

#uma otima maneira de ler arquivos GeoTiff no R eh usando o pacote "rgdal"
#instale o pacote "rgdal" no seu R "install.packages(rgdal,dependencies=T)"

?GDALinfo #com o help, aprenda a utilizar o comando!

#carregue o pacote "require(rgdal)"
?GDALinfo #agora sim o pacote esta carregado!

GDALinfo("./imagem/B4.tif") #informacoes sobre o arquivo GeoTiff

B4.con<-GDAL.open("./imagem/B4.tif") #cria uma conexao. o dado nao esta no workspace mas
pode ser acessado
displayDataset(B4.con) #plota a imagem
dim(B4.con) #mostra as quantas linhas e colunas tem a imagem

#uma maneira de acessar os valores dos pixels e converte-los para a forma matricial eh fazendo
B4.matriz<-getRasterData(B4.con) #le no formato de matriz
image(B4.matriz) #desenhando a imagem

```

```
dim(B4.matriz) #observe que ao ler, ele o faz da na forma transposta

#corrigindo o problema da transposta
B4.matriz<-t(getRasterData(B4.con)) #le no formato de matriz
dim(B4.matriz) #agora sim!!!
B4.matriz[300:310,400:410] #acessa parte da imagem

GDAL.close(B4.con) #fecha a conexao, mas nao exclui o objeto "B4.con"

B4.file<-readGDAL("./imagem/B4.tif") #le o GeoTiff
summary(B4.file) #informacoes sobre o objeto "B4.file"
```