

Wataru

Proposta de trabalho final

Os objetivos do programa são: 1- organizar as informações contidas em uma tabela produzida por um computador que registra as informações de uma tarefa envolvendo ratos. Cada linha da tabela representa uma tentativa. As colunas representam: tentativa, lado da apresentação do estímulo 1 (direita ou esquerda), lado da apresentação do estímulo 2 (direita ou esquerda), intervalo entre as apresentações dos estímulos (100, 400 e 600ms), tempo de resposta do animal ao estímulo e outras (sem importância para as minhas análises). As combinações entre os lados de apresentação dos estímulos será resumido em válidas (estímulo 1 e 2 aparecem do mesmo lado) e inválidas (estímulos 1 e 2 aparecem em lados opostos) Assim, os dados de saída para cada tabela será a mediana do tempo de reação em: válida 100, inválida 100, válida 400, inválida 400 e válida 600, inválida 600. Cada um dos arquivos de saída pode conter registros de mais de uma sessão; para cada sessão será produzida um conjunto de dados de saída.

Ao final do processo será produzida uma matriz com as medianas (ou média das medianas caso exista mais de uma sessão por rato) de cada animal.

2- em seguida o programa construirá um ou mais gráficos a partir das informações contidas na matriz de saída.

3- finalmente, com a matriz de saída, o programa realizará o teste de análise de variância comparando tempo de reação entre tentativas válidas e inválidas em cada um dos intervalos.

DADOS

Conjunto de dados no formato para utilização da função

1 2 3 4 5 6 7 8 9

Código Trabalho Final

```
analise.geral=function(comprimento, sessao){
  saidafinal=matrix(NA,nrow = length(dir(pattern=".DAT")), ncol = 6, byrow
= T)# cria matriz linhas=numero de arquivos
  compfinal=matrix(NA,nrow = length(dir(pattern=".DAT")), ncol = 6, byrow
= T) # idem
  for(i in 1:length(dir(pattern=".DAT"))){
cada uma das iterações lê um dos arquivos      ##
```

```
tabela=read.table(dir(pattern=".DAT")[i], sep="\t", dec=".")
#"tabela" recebe dados do arquivo lido
tabela$V12=as.numeric(paste(tabela$V2,tabela$V4, sep=""))
#insere no data.frame combinação de V2(estimulo 1) e V4(estimulo 2)
tabela$V13=rep(seq(from=1,to=(length(tabela$V1)/comprimento)),
each=comprimento) # insere em cada linha da tabela a sessao ao qual ela
pertence
saida1=saida2=saida3=saida4=saida5=saida6=NA
#cria vetores
comp1=comp2=comp3=comp4=comp5=comp6=NA
#idem
for(j in 1 :(length(tabela$V1)/comprimento)){
##de 1 a numero de sessoes, calcula as medianas das condicoes:
saida1[j]=median(tabela$V10[tabela$V9==10&tabela$V8==100&(tabela$V12==321|ta
bela$V12==644)&tabela$V13==j]) #tempo de reação(V10)das
###tentativas corretas(V9=10)das tentativas
validas(V12=321 ou 644) no SOA 100ms (V8=100) da corrente sessao(V13=j)
saida2[j]=median(tabela$V10[tabela$V9==10&tabela$V8==100&(tabela$V12==324|ta
bela$V12==641)&tabela$V13==j])# invalida soa 100
saida3[j]=median(tabela$V10[tabela$V9==10&tabela$V8==400&(tabela$V12==321|ta
bela$V12==644)&tabela$V13==j])# valida soa 400
saida4[j]=median(tabela$V10[tabela$V9==10&tabela$V8==400&(tabela$V12==324|ta
bela$V12==641)&tabela$V13==j])#...
saida5[j]=median(tabela$V10[tabela$V9==10&tabela$V8==600&(tabela$V12==321|ta
bela$V12==644)&tabela$V13==j])
saida6[j]=median(tabela$V10[tabela$V9==10&tabela$V8==600&(tabela$V12==324|ta
bela$V12==641)&tabela$V13==j])
}
if(sessao>0){ #calcula uma sessao isoladamente
saidafinal[i,1]=saida1[sessao]
saidafinal[i,2]=saida2[sessao]
saidafinal[i,3]=saida3[sessao]
saidafinal[i,4]=saida4[sessao]
saidafinal[i,5]=saida5[sessao]
saidafinal[i,6]=saida6[sessao]
}
else{ #calcula com base na media das
sessoes
saidafinal[i,1]=mean(na.omit(saida1))
saidafinal[i,2]=mean(na.omit(saida2))
saidafinal[i,3]=mean(na.omit(saida3))
saidafinal[i,4]=mean(na.omit(saida4))
saidafinal[i,5]=mean(na.omit(saida5))
saidafinal[i,6]=mean(na.omit(saida6))
}
for(l in 1 :(length(tabela$V1)/comprimento)){ ##calcula o numeros
de acertos em cada condicao
comp1[l]=length(tabela$V10[tabela$V9==10&tabela$V8==100&(tabela$V12==321|tab
ela$V12==644)&tabela$V13==l])
comp2[l]=length(tabela$V10[tabela$V9==10&tabela$V8==100&(tabela$V12==324|tab
```

```

ela$V12==641)&tabela$V13==l])
comp3[l]=length(tabela$V10[tabela$V9==10&tabela$V8==400&(tabela$V12==321|tab
ela$V12==644)&tabela$V13==l])
comp4[l]=length(tabela$V10[tabela$V9==10&tabela$V8==400&(tabela$V12==324|tab
ela$V12==641)&tabela$V13==l])
comp5[l]=length(tabela$V10[tabela$V9==10&tabela$V8==600&(tabela$V12==321|tab
ela$V12==644)&tabela$V13==l])
comp6[l]=length(tabela$V10[tabela$V9==10&tabela$V8==600&(tabela$V12==324|tab
ela$V12==641)&tabela$V13==l])
    }
    if(sessao>0){          ##calcula uma sessao isoladamente
compfinal[i,1]=comp1[sessao]/length(tabela$V10[tabela$V8==100&(tabela$V12==3
21|tabela$V12==644)&tabela$V13==l])
compfinal[i,2]=comp2[sessao]/length(tabela$V10[tabela$V8==100&(tabela$V12==3
24|tabela$V12==641)&tabela$V13==l])
compfinal[i,3]=comp3[sessao]/length(tabela$V10[tabela$V8==400&(tabela$V12==3
21|tabela$V12==644)&tabela$V13==l])
compfinal[i,4]=comp4[sessao]/length(tabela$V10[tabela$V8==400&(tabela$V12==3
24|tabela$V12==641)&tabela$V13==l])
compfinal[i,5]=comp5[sessao]/length(tabela$V10[tabela$V8==600&(tabela$V12==3
21|tabela$V12==644)&tabela$V13==l])
compfinal[i,6]=comp6[sessao]/length(tabela$V10[tabela$V8==600&(tabela$V12==3
24|tabela$V12==641)&tabela$V13==l])
    }
    else{          ##calcula com base na media das
sessoes
compfinal[i,1]=mean(na.omit(comp1))/length(tabela$V10[tabela$V8==100&(tabela
$V12==321|tabela$V12==644)&tabela$V13==l])
compfinal[i,2]=mean(na.omit(comp2))/length(tabela$V10[tabela$V8==100&(tabela
$V12==324|tabela$V12==641)&tabela$V13==l])
compfinal[i,3]=mean(na.omit(comp3))/length(tabela$V10[tabela$V8==400&(tabela
$V12==321|tabela$V12==644)&tabela$V13==l])
compfinal[i,4]=mean(na.omit(comp4))/length(tabela$V10[tabela$V8==400&(tabela
$V12==324|tabela$V12==641)&tabela$V13==l])
compfinal[i,5]=mean(na.omit(comp5))/length(tabela$V10[tabela$V8==600&(tabela
$V12==321|tabela$V12==644)&tabela$V13==l])
compfinal[i,6]=mean(na.omit(comp6))/length(tabela$V10[tabela$V8==600&(tabela
$V12==324|tabela$V12==641)&tabela$V13==l])
    }
}
cont1=col=1          #contadores
v=i=vp=ip=soa=NA    #cria vetores
for(t in 1:3){      #3 = numero de SOAs(100, 400, 600)
  for(u in 1:length(dir(pattern=".DAT"))){ # para cada uma das
linhas das matrizes "saidafinal" e "compfinal"
    if(compfinal[u,col+1]!="NaN"){ # copia apenas quando ha
tentativas invalidas pareando com uma tentativa valida
      v[cont1]=saidafinal[u,col] ## grava em vetores separados o
tempo de reacao (val e inv) e a porcentagem de acertos e o SOA
      i[cont1]=saidafinal[u,col+1]
      vp[cont1]=compfinal[u,col]
    }
  }
}

```

```
        ip[cont1]=compfinal[u,col+1]
        soa[cont1]=col
        cont1=cont1+1      ## soma 1 ao contador que determina a
posicao de gravacao dos dados nos vetores
    }
}
col=col+2                ## soma 2 ao contador para testar apenas as
colunas da matriz com tentativas invalidas
}
resumo=data.frame(c(v,i),c(vp,ip),c(rep("val", length(v)), rep("inv",
length(i))),c(soa,soa)) ## constroi um unico data frame contendo o TR e
porcentagem de acertos indexados por validade e pelo SOA
names(resumo)= c("TR","Porcent", "Validade", "SOA")

### parte grafica

library(graphics)
par(mfrow=c(1,2))

#constroi grafico para o tempo de reacao
interaction.plot(resumo$SOA,resumo$Validade,resumo$TR,fun=mean,ylab="Média
Tempo de Reação", xlab="SOA", trace.label="Validade", ylim=c(250,500))
# o calculo das medias e dos desvios considera a existencia de ate tres
soas
media=c(mean(resumo$TR[resumo$SOA=="1"&resumo$Validade=="val"]),mean(resumo$
TR[resumo$SOA=="1"&resumo$Validade=="inv"]),
mean(resumo$TR[resumo$SOA=="3"&resumo$Validade=="val"]),mean(resumo$TR[resum
o$SOA=="3"&resumo$Validade=="inv"]),
mean(resumo$TR[resumo$SOA=="5"&resumo$Validade=="val"]),mean(resumo$TR[resum
o$SOA=="5"&resumo$Validade=="inv"]))
mediaf=na.omit(media) # remove "NaN caso os dados analisados contenham
apenas 2 ou 1 soa
desv=c(sd(resumo$TR[resumo$SOA=="1"&resumo$Validade=="val"]),sd(resumo$TR[re
sumo$SOA=="1"&resumo$Validade=="inv"]),
sd(resumo$TR[resumo$SOA=="3"&resumo$Validade=="val"]),sd(resumo$TR[resumo$SO
A=="3"&resumo$Validade=="inv"]),
sd(resumo$TR[resumo$SOA=="5"&resumo$Validade=="val"]),sd(resumo$TR[resumo$SO
A=="5"&resumo$Validade=="inv"]))
desvf=na.omit(desv) # remove "NaN caso os dados analisados contenham
apenas 2 ou 1 soa
arrows(rep(1:(length(mediaf))/2),
each=2),mediaf[1:length(mediaf)]+desvf[1:length(desvf)], #desenha as barras
de desvio no grafico
rep(1:(length(mediaf))/2), each=2),mediaf[1:length(mediaf)]-
desvf[1:length(desvf)],code=3,angle=90)

#constroi grafico para porcentagem de acertos...idem ao anterior
interaction.plot(resumo$SOA,resumo$Validade,resumo$Porcent,fun=mean,ylab="Mé
dia porcentagem de acertos", xlab="SOA",
trace.label="Validade",ylim=c(0.2,0.5))
```

```

mediap=c(mean(resumo$Porcent[resumo$SOA=="1"&resumo$Validade=="val"]),mean(r
esumo$Porcent[resumo$SOA=="1"&resumo$Validade=="inv"]),
mean(resumo$Porcent[resumo$SOA=="3"&resumo$Validade=="val"]),mean(resumo$Por
cent[resumo$SOA=="3"&resumo$Validade=="inv"]),
mean(resumo$Porcent[resumo$SOA=="5"&resumo$Validade=="val"]),mean(resumo$Por
cent[resumo$SOA=="5"&resumo$Validade=="inv"]))
  mediafp=na.omit(mediap)
desvp=c(sd(resumo$Porcent[resumo$SOA=="1"&resumo$Validade=="val"]),sd(resumo
$Porcent[resumo$SOA=="1"&resumo$Validade=="inv"]),
sd(resumo$Porcent[resumo$SOA=="3"&resumo$Validade=="val"]),sd(resumo$Porcent
[resumo$SOA=="3"&resumo$Validade=="inv"]),
sd(resumo$Porcent[resumo$SOA=="5"&resumo$Validade=="val"]),sd(resumo$Porcent
[resumo$SOA=="5"&resumo$Validade=="inv"]))
  desvfp=na.omit(desvp)
  arrows(rep(1:((length(mediafp))/2),
each=2),mediafp[1:length(mediafp)]+desvfp[1:length(desvfp)],
        rep(1:((length(mediafp))/2), each=2),mediafp[1:length(mediafp)]-
desvfp[1:length(desvfp)],code=3,angle=90)
  par(mfrow=c(1,1))

  Anova.TR=summary(aov(resumo$TR~resumo$Validade*resumo$SOA))
#teste anova para tempo de reacao
  Anova.Pc=summary(aov(resumo$Porcent~resumo$Validade*resumo$SOA))
#teste anova para porcentagem de acertos
  return(resumo, Anova.TR, Anova.Pc) #retorna
tabela com as medianas do tempo de reacao nas tentativas com pista valida e
invalida,
                                     #a porcentagem de
acertos para tentativas validadas e invalidas, um indexador para a validade
e outro para o soa
}

```

Help

analise.geral

package:unknown

R Documentation

```
analise.geral(comprimento, sessao)
```

Description:

A funcao retorna a análise dos dados obtidos pelo programa 090327 produzido em nosso laboratorio para coleta de dados em uma tarefa para avaliar a atencao em ratos. Lê todos os arquivos gravados em uma pasta e retorna uma tabela resumida, representação gráfica e os resultados de testes estatísticos.

Usage:

funcao(comprimento, sessao)

Arguments:

comprimento inteiro representado numeros total de tentativas por sessao
sessao a sessao de interesse para a análise (e.g. 1[1ª sessao], 2[2ª sessao], etc.) sessao=0 retorna a media das sessões

Details:

Não há argumento para carregar arquivo de dados. Isso é feito automaticamente pelo programa que lê os arquivo com extensão ".DAT" ou como parte do nome contidos na mesma pasta onde está salvo a área de trabalho do R.

Não deve haver uma linha de cabeçalho ou separações entres as sessões(se houver mais de uma sessão).

o formato do arquivo deve ser:

coluna 2 : estímulo 1 (pista) (32 ou 64)

coluna 4 : estímulo 2 (alvo) (1 ou 4)

coluna 8 : SOA (100, 400 ou 600)

coluna 9 : Tipo de erro (1 a 10)

coluna 10 : Tempo de reação(ms)

As demais colunas não são utilizadas para fins de análise nesse programa.

Value:

A função retorna:

1: tabela com as medianas do tempo de reação e porcentagem de acertos indexadas por validade ("val" e "inv") e SOA.

2: sumário do teste análise de variancia do tempo de reação em função de validade e SOA

3: sumário do teste análise de variancia da porcentagem de acertos em função de validade e SOA

Author(s):

Wataru Sumi

Examples:

```
##utilize setwd(endereço da pasta onde estão os arquivos a serem analisados)
```

```
##para os dados utilizados como exemplo:
```

```
analise.geral(comprimento=480, sessao=0)
```

From:

<http://ecor.ib.usp.br/> - **ecoR**

Permanent link:

http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2010:alunos:trabalho_final:wataru.sumi:start 

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