

Mario José Marques Azevedo



Aluno de mestrado em ecologia

Departamento de Botânica, Instituto de Biologia, UNICAMP.

Linha de pesquisa: Modelos de distribuição de abundância para árvores de Floresta Estacional Semidecídua e Cerradão

Orientadores: Prof. Dr. Fernando Roberto Martins (Unicamp) e Dr. Roque Cielo Filho (Instituto Florestal)

Meus exercícios

Página com meus [exercícios](#) resolvidos.

Proposta de trabalho final

Proposta A

Analisar qual a distribuição dos dados criando a própria distribuição de probabilidade. A função receberá um vetor de dados e irá verificar qual a distribuição da média deste conjunto de dados. Para isso será utilizado algumas distribuições disponíveis no R para gerar as distribuições de probabilidade teórica e calcular a significância dos dados frente a cada distribuição (a exemplo da aula de inferência do Prof. Alexandre). Este teste serve como teste de normalidade dos dados, quando comparado com a distribuição normal teórica gerada. A saída será uma tabela com os valores de p para cada distribuição.

Proposta B

Fazer um para testar hipótese para duas amostras. A entrada seria dois vetores de dados. A normalidade e variância dos vetores seriam analisadas para saber qual teste usar. Se os dados forem normais com variâncias iguais será realizado um teste t padrão. Se as variâncias forem diferentes será utilizado um teste t com correção de Welch. Se os dados não satisfizerem estas premissas, um teste de Wilcoxon será realizado.

A função terá como parâmetros básicos a entrada dos dois vetores. Como parâmetros adicionais será considerado como devem ser tratados os NAs e se o teste será bicaudal ou unicaudal.

A saída apresentará o teste utilizado para comparação com sua respectiva estatística e p-valores tanto para teste unicaudal quando bicaudal, quando não for explicitamente declarado a preferência.

Serão utilizadas as próprias funções implementadas no R para o cálculo dos testes. O script será para manipular os dados e verificar qual teste melhor se adequa aos dados.

Comentários da proposta (Leo)

Considero as duas propostas úteis e factíveis. Particularmente gosto mais da proposta A. Neste caso, tu estará testando a distribuição para variáveis da classe numérica. Acho interessante ressaltar isto porque os dados podem seguir outras distribuições, sendo de natureza binária (0 ou 1), probabilística (binomial de 0 a 1), poisson (inteiros positivos), e etc. Então, uma complicação da sua função poderia ser você determinar qual a classe do vetor que será testado, como um argumento da função. Não precisa ir tão longe, mas caso seja fácil atingir a tua proposta, sugiro que você tente implementar outros argumentos. Outra coisa que considero útil, que pode ser um dos outputs da função, seria algum diagnóstico gráfico. Neste caso, você poderia plotar as curvas dos dados e das diferentes distribuições. Outra ideia seria algum diagnóstico de assimetria da distribuição (p.ex. curtose).

Página de ajuda

`duas.amostras.teste`
R Documentation

`package:nenhum`

Teste de hipótese a respeito de duas amostras verificando antes as premissas do teste.

Description:

Verifica se os dados satisfazem os pressupostos dos Teste t de Student e Wilcoxon (ou Mann-Whitney) para decidir qual destes usar.

Usage:

```
duas.amostras.teste(vector.a, vector.b, ...)  
## Default method:  
duas.amostras.teste(vector.a, vector.b, alt.hypothesis="two.sided",  
signif.level=0.05, paired=FALSE,  
mu=0, w.correct=TRUE, removeNA=TRUE,  
return.htest=FALSE)
```

Arguments:

<code>vector.a</code> e <code>vector.b</code>	vetores numéricos (não vazios) com os dados.
<code>alt.hypothesis</code>	string de caractere especificando a hipótese alternativa. Pode ser "two.sided" (default), "greater" ou "less" (ver em Detalhes).
<code>signif.level</code>	valor numérico do nível de significância do teste.
<code>paired</code>	valor lógico indicando se deseja teste pareado (ver em Detalhes).
<code>mu</code>	valor numérico indicando o parâmetro da hipótese nula.
<code>w.correct</code>	valor lógico indicando se deve ser aplicado a

correção para normalidade para o valor
de p no teste de Wilcoxon.
removeNA valor lógico indicando se os NAs devem ser
removidos. Caso não seja, NAs serão
convertidos em zeros.
return.htest valor lógico indicando se a função deve retornar o
objeto de classe 'htest' do teste
ou imprimir o resultado na tela.

Details:

A função é aplicável somente para testes com duas amostras. Os vetores passados como parâmetros serão analisados para verificar se são do tipo 'numeric' ou 'integer'. A presença de NAs será verificada. Se for especificado o parâmetro removeNA=TRUE, os NAs serão removidos e vetores menores, sem os NAs serão analisados. Caso o parâmetro seja falso, os NAs serão convertidos em zeros. O tamanho dos vetores serão verificados. Vetores de tamanho 1 ou nulos retornarão erro. Quando o parâmetro paired=TRUE, os vetores devem ser do mesmo tamanho.

A função verifica se os dados são normais para escolher entre o Teste t e o Teste de Wilcoxon. Caso os dados sejam normais, a função escolhe o primeiro teste, senão escolhe o segundo. Posteriormente é verificada a homoscedasticidade dos dados. Para o Teste t quando as variâncias são diferentes, a correção de Welch é aplicada ao grau de liberdade.

Uma das premissas do Teste de Wilcoxon é a igualdade das variâncias (Zar 2010). Quando o tamanho das amostras não são iguais e a maior variância está associada a maior amostra, a probabilidade do Erro Tipo I será menor que o especificado no nível de significância (Zar 2010). Se a maior variância está associada a menor amostra, a probabilidade do Erro Tipo I é maior que a especificada no nível de significância (Zar 2010).

O parâmetro alt.hypothesis define a hipótese alternativa. Para o Teste t, a diferença entre as médias analisadas para a hipótese alternativa pode ser: diferente do parâmetro μ , por padrão igual a zero, para a análise bicaudal (two.sided); menor que μ (less) ou maior que μ (greater) para testes unicaudais.

Para o Teste de Wilcoxon a hipótese alternativa pode ser: a localização (média ou mediana) da distribuição dos dados dos vetores 'a' e 'b' são diferentes de μ (two.sided); a localização do vetor 'a' maior que a do vetor 'b' (greater) ou a localização do vetor 'a' menor que a do vetor 'b' (less).

Value:

Lista com a classe "htest" contendo os seguintes componentes (return.htest=TRUE).

statistic valor da estatística do teste.
parameter graus de liberdade (Teste t) ou parâmetros da distribuição (Teste de Wilcoxon).
p.value valor de p para o teste.
null.value valor hipotético da média (Teste T) ou desvio dos valores (Teste de Wilcoxon) da hipótese nula.
alternative string descrevendo a hipótese alternativa.
method descrição do tipo do teste utilizado.
data.name string de caractere com o nome dos dados utilizados.
conf.int intervalo de confiança.
estimate valor estimado da diferença das médias (Teste t) ou desvio dos dados (Teste de Wilcoxon).

Author(s):

Mario José Marques Azevedo
Estudante de Mestrado em Ecologia. Universidade Estadual de Campinas, Campinas, Brasil.

References:

Hollander and Douglas A. Wolfe (1973), Nonparametric Statistical Methods. New York: John Wiley & Sons.
Pages 27–33 (one-sample), 68–75 (two-sample). Or second edition (1999).

Jarrold H. Zar, J. 2010. Biostatistical analysis. 5^o ed., Prentice Hall, London.

Note:

Requer pacote 'car'

See Also:

t.test(), wilcox.test() e var.test() do pacote 'stats' e leveneTest() do pacote 'car'.

Examples:

```
# Zar 2010 Biostatistical Analysis. Example 8.1 two tailed t test
# Blood-Clotting Times (in Minutes) of Male Adult Rabbits Given One of
Two Different Drugs
# Answer: Mean blood-clotting time is not the same for subjects
receiving drug B as it is for
# receiving drug G.
```

```
example8.1.a<-c(8.8,8.4,7.9,8.7,9.1,9.6); # Drug B
example8.1.b<-c(9.9,9.0,11.1,9.6,8.7,10.4,9.5); # Drug G
duas.amostras.teste(example8.1.a,example8.1.b);

# Zar 2010 Biostatistical Analysis. Example 8.2 one tailed t test
# Heights of Plants, Each Grown with One of Two Different Fertilizers
# Answer: The mean plant height (in cm) is greater with the newer
fertilizer.
example8.2.a<-c(48.2,54.6,58.3,47.8,51.4,52.0,55.2,49.1,49.9,52.6); #
Present fertilizer
example8.2.b<-c(52.3,57.4,55.6,53.2,61.3,58.0,59.8,54.8); # Newer
fertilizer
duas.amostras.teste(example8.2.a,example8.2.b,alt.hypothesis="less");

# Zar 2010 Biostatistical Analysis. Example 8.2a two tailed t test with
Welch correction
# Times for seven cockroach eggs to hatch at one laboratory
temperature and for eight eggs to hatch at
# another temperature
# Answer: Means of temperature are different
example8.2a.a<-c(40,38,32,37,39,41,35); # At 30°C
example8.2a.b<-c(36,45,32,52,59,41,48,55); # At 10°C
duas.amostras.teste(example8.2a.a,example8.2a.b);

# Zar 2010 Biostatistical Analysis. Example 9.1 two tailed paired t test
# Measure of forelegs and hindlegs lengths of deer
# Answer: Mean of differences between hindlegs and forelegs are
different 0
example9.1.a<-c(142,140,144,144,142,146,149,150,142,148); # Hindleg
length (cm)
example9.1.b<-c(138,136,147,139,143,141,143,145,136,146); # Foreleg
length (cm)
duas.amostras.teste(example9.1.a,example9.1.b,paired=TRUE);

# Zar 2010 Biostatistical Analysis. Example 9.2 one tailed paired t test
# Test whether a new fertilizer results in an increase of more than
250 kg/ha in crop yield over the
# old fertilizer
# Answer: New fertilizer not increase of more than 250 kg/ha in crop
yield
example9.2.a<-c(2250,2410,2260,2200,2360,2320,2240,2300,2090); # With
new fertilizer
example9.2.b<-c(1920,2020,2060,1960,1960,2140,1980,1940,1790); # With
old fertilizer
duas.amostras.teste(example9.2.a,example9.2.b,paired=TRUE,alt.hypothesis="gr
eater",mu=250);

# Hollander & Wolfe (1973), 29f. One tailed paired Wilcoxon test
# Hamilton depression scale factor measurements in 9 patients with
mixed anxiety and depression, taken
```

```
# at the first (x) and second (y) visit after initiation of a
therapy (administration of a
# tranquilizer)
x <- c(1.83,0.50,1.62,2.48,1.68,1.88,1.55,3.06,1.30)
y <- c(0.878,0.647,0.598,2.05,1.06,1.29,1.06,3.14,1.29)
duas.amostras.teste(x,y,paired=TRUE,alt.hypothesis="greater")
# Myles Hollander and Douglas A. Wolfe (1973), Nonparametric Statistical
Methods. New York: John Wiley &
# Sons. Pages 27–33 (one-sample), 68–75 (two-sample). Or second
edition (1999).
```

```
# Teste de Wilcoxon bicaudal com violacao da premissa de igualdade de
variancias
```

```
set.seed(42);
x<-runif(30,1,30);
y<-runif(35,1,60);
duas.amostras.teste(x,y);
```

```
# Teste de Wilcoxon bicaudal com correcao para normalidade
```

```
set.seed(42);
x<-runif(30,1,30);
y<-runif(50,1,30);
duas.amostras.teste(x,y);
```

```
# Teste de Wilcoxon bicaudal com correcao para normalidade
```

```
set.seed(42);
x<-c(rnorm(30),NA,NA);
y<-c(rnorm(50),NA);
duas.amostras.teste(x,y);
```

Código da função

```
# Proposta B
#
# Verifica premissas de testes de hipotese para duas amostras para decidir
qual teste utilizar
# Autor: Mario Jose Marques
# email: mariojosebr [at] yahoo [dot] com [dot] br
# Data: 18 abr 2012
# Versao: 1

duas.amostras.teste<-
function(vector.a=NULL,vector.b=NULL,alt.hypothesis="two.sided",signif.level
=0.05,paired=FALSE,
mu=0,w.correct=TRUE,removeNA=TRUE,return.htest=FALSE,conf.int=TRUE){
  # Verifica e carrega o pacote car para utilizar o teste de Levene para
  variancias
  if(!require("car")){
```

```
stop("Pacote 'car' necessário não instalado\n",call.=FALSE);
}
# Verifica classe dos vetores
if (!is.numeric(vector.a) & !is.integer(vector.a)){
  stop("Vetor 'a' nao e do tipo 'numeric' ou 'integer'\n",call.=FALSE);
}
if (!is.numeric(vector.b) & !is.integer(vector.b)){
  stop("Vetor 'b' nao e do tipo 'numeric' ou 'integer'\n",call.=FALSE);
}
# Manipula NAs
if(sum(is.na(vector.a))>0 | sum(is.na(vector.b))>0){
  if(removeNA==TRUE){
    cat("\n\tObs.:",sum(is.na(vector.a))>0, is.na(vector.b)>0),"NA
removidos\n");
    vector.a<-vector.a[!is.na(vector.a)];
    vector.b<-vector.b[!is.na(vector.b)];
  } else {
    cat("\n\tObs.:",sum(is.na(vector.a))>0, is.na(vector.b)>0),"NA
convertidos para 0\n");
    vector.a[is.na(vector.a)]<-0;
    vector.b[is.na(vector.b)]<-0;
  }
}
# Verifica o tamanho dos vetores
if(length(vector.a)<=1){
  stop("Tamanho do vetor 'a' insuficiente\n",call.=FALSE);
}
if(length(vector.b)<=1){
  stop("Tamanho do vetor 'b' insuficiente\n",call.=FALSE);
}
if(paired==TRUE & (length(vector.a)!=length(vector.b))){
  stop("Tamanhos de vetores diferentes para teste de amostras
pareadas\n",call.=FALSE);
}

### Verifica a normalidade dos dados
shap.a<-shapiro.test(vector.a);
shap.b<-shapiro.test(vector.b);
### Verifica a igualdade de variancia dos dados
if(shap.a$p.value>=signif.level & shap.b$p.value>=signif.level){
  # Se os dados forem normais, utiliza o teste da razao de variacao de
Fisher.
  normality<-TRUE;
  vartest<-var.test(vector.a,vector.b);
  ifelse(vartest$p.value>=signif.level, equalvar<-TRUE, equalvar<-FALSE);
} else {
  # Se ao menos um dos vetores nao for normal, realiza o teste de Levene
(pacote car).
  normality<-FALSE;
  temp<-
data.frame(measure=c(vector.a,vector.b),group=c(rep("a",length(vector.a)),re
```

```
p("b",length(vector.b))));  
  temp$group<-as.factor(temp$group);  
  vartest<-leveneTest(measure~group,data=temp);  
  ifelse(vartest[["Pr(>F)"]][1]>=signif.level, equalvar<-TRUE, equalvar<-  
FALSE);  
}  
  
### Realiza o teste de hipotese para o conjunto de dados  
if(normality==TRUE){  
  # Realiza do test t  
  ttest<-  
t.test(vector.a,vector.b,mu=mu,alternative=alt.hypothesis,paired=paired,var.  
equal=equalvar);  
  # Verifica se retorna objeto htest ou imprime informacoes na tela  
  if(return.htest==TRUE){  
    return(ttest);  
  } else {  
    # Imprime o nome do teste  
    cat("\n\t Teste t ");  
    if(paired==TRUE){  
      cat("pareado\n");  
    } else {  
      cat("para duas amostras");  
      if(equalvar==FALSE){  
        cat(" com correcao de Welch\n");  
      } else {  
        cat("\n");  
      }  
    }  
    # Imprime as hipoteses do teste  
    if(alt.hypothesis=="greater"){  
      cat("Teste unicaudal:\n\tH0: diferenca das medias dos vetores 'a' e  
'b' menor ou igual a",mu,  
        "\n\tH1: diferenca das medias dos vetores maior que",mu,"\n");  
    }  
    else if (alt.hypothesis=="less"){  
      cat("Teste unicaudal:\n\tH0: diferenca das medias dos vetores 'a' e  
'b' maior ou igual a",mu,  
        "\n\tH1: diferenca das medias dos vetores menor que",mu,"\n");  
    }  
    else if (alt.hypothesis=="two.sided"){  
      cat("Teste bicaudal:\n\tH0: diferenca das medias dos vetores 'a' e  
'b' igual a",mu,  
        "\n\tH1: diferenca das medias dos vetores diferente  
de",mu,"\n");  
    }  
    # Imprime a estatistica do teste  
    cat("t =",ttest$statistic,"\nGrau de liberdade  
=",ttest$parameter,"\nValor de p =",ttest$p.value,"\n");  
    if(paired==TRUE){
```

```
        cat("Media da diferenca =",ttest$estimate[[1]],"\n");
    } else {
        cat("Media do vetor 'a' =",ttest$estimate[[1]],"\nMedia do vetor 'b'
=",ttest$estimate[[2]],"\n");
    }
    # Imprime a interpretacao do teste ao nivel de significancia
    especificado
    if (ttest$p.value>=signif.level) { cat("Nao ha "); } else { cat ("Ha
"); }
    cat("evidencias, ao nivel de significancia de ",signif.level*100,"%,
para rejeitar H0",sep="");
    } # Fim da condicao para retorno do objeto htest
} else {
    # Realiza do teste de Wilcoxon
    utest<-
wilcox.test(vector.a,vector.b,mu=mu,alternative=alt.hypothesis,paired=paired
,correct=w.correct,conf.int=conf.int);

    # Verifica se retorna objeto htest ou imprime informacoes na tela
    if(return.htest==TRUE){
        return(utest);
    } else {
        # Imprime o nome do teste
        cat("\n\tTeste de Wilcoxon ");
        if(paired==TRUE){
            cat("pareado");
        } else {
            cat("para duas amostras");
        }
        if(w.correct==TRUE & (length(vector.a)>=50 | length(vector.b)>=50)){
            cat(" com aproximacao para normal\n");
        } else {
            cat("\n");
        }
        # Imprime a estatistica do teste
        cat("W =",utest$statistic,"\nValor de p =",utest$p.value,"\n");
        # Imprime as hipoteses do teste
        if(alt.hypothesis=="greater"){
            cat("Teste unicaudal:\n\tH0: a localizacao (media ou mediana) do
vetor 'a' em relacao ao 'b' menor ou igual a",mu,
            "\n\tH1: a localizacao do vetor 'a' em relacao ao 'b' maior
que",mu,"\n");
        }
        else if (alt.hypothesis=="less"){
            cat("Teste unicaudal:\n\tH0: a localizacao (media ou mediana) do
vetor 'a' em relacao ao 'b' maior ou igual a",mu,
            "\n\tH1: a localizacao do vetor 'a' em relacao ao 'b' menor
que",mu,"\n");
        }
        else if (alt.hypothesis=="two.sided"){
            cat("Teste bicaudal:\n\tH0: a localizacao (media ou mediana) dos
```

```
valores dos vetores 'a' e 'b' sao iguais a",mu,
"\n\tH1: a localizacao dos valores dos vetores sao diferentes
de",mu,"\n");
}
# Imprime a interpretacao do teste ao nivel de significancia
especificado
if (utest$p.value>=signif.level) { cat("Nao ha "); } else { cat ("Ha
"); }
cat("evidencias, ao nivel de significancia de ",signif.level*100,"%
para rejeitar H0\n",sep="");

} # Fim da condicao para retornar objeto htest
# Avisa do vies do teste quando as variancias nao sao iguais
if (equalvar==FALSE){
cat("Obs.: Variancias nao sao iguais");
if((length(vector.a)>length(vector.b) & var(vector.a)>var(vector.b)) |
(length(vector.a)<length(vector.b) & var(vector.a)<var(vector.b))){
cat(": Erro tipo I menor que", signif.level,"\n");
}
if((length(vector.a)>length(vector.b) & var(vector.a)<var(vector.b)) |
(length(vector.a)<length(vector.b) & var(vector.a)>var(vector.b))){
cat(": Erro tipo I maior que", signif.level,"\n");
}
}
} # Fim da condicao do teste para variavel normality
} # Fim da funcao
```

Arquivo da função

Clique [aqui](#) para baixar o arquivo da função e [aqui](#) para baixar o arquivo de ajuda.

From:

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

Permanent link:

http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:alunos2012:alunos:trabalho_final:mariojosebr:start

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

