

## André Ramos Vaquero



Doutorando em Ciências Médicas, Faculdade de Medicina, USP.

O título de minha tese é: “Comparação de redes gênicas para o entendimento de efeitos tóxicos de glicose e de variantes genéticas no risco de Diabetes Mellitus tipo 2”, orientado por Alexandre da Costa Pereira.

### MEUS EXERCÍCIOS [exec](#)

### Proposta de Trabalho Final

#### Principal

A função tem a intenção de identificar diferenças entre médias de variáveis quantitativas de dois grupos, onde é inserida uma tabela com dados qualitativos (1 coluna, com 2 grupos) e quantitativos (1 coluna ou mais). Anteriormente ao teste de diferença de médias, serão feitos testes de normalidade e variancia para decidir que tipo de análise será feita. Se houver normalidade nos dados das colunas (todos indivíduos juntos) e se houver normalidade dos dados intra-grupo e variancias iguais entre os grupos será feito um teste T, se esses requisitos não forem atendidos será realizado um teste de comparação de médias por permutação. Para cada coluna serao feitos graficos comparativos.

#### [Help da função](#)

dia 10 de abril , nessa funcao so consegui fazer um teste t

```
t.test.tabelas=function(tab,tab1) #insira aqui suas tabelas, que contem o
mesmo numero de colunas, com dados #quantitativos, a coluna 1 das duas
tabelas terao suas medias comparadas através de um teste t, isso se
repetindo #para as demais colunas seguintes
{
  if(ncol(tab)==ncol(tab1)) #se as tabelas tiverem o mesmo numero de
colunas, o programa segue, se nao vai para o #else
  {
    dim.tab=dim(tab)[2] #criei um vetor para dimensao,numero de
colunas, da tabela 1
    dim.tab1=dim(tab1)[2] #criei um vetor para dimensao,numero de
colunas, da tabela 2
    testes=c(0) #criei um vetor para resultado final
    for (i in 1:(dim.tab)) #criei um contador começando em 1 e
terminando na dimensao das tabelas
    {
      testes[i]= t.test(tab[,i],tab1[,i])[3] #realiza o teste t,
para colunas com numeros iguais de cada #tabela, e retira a terceira posicao
do teste colocando-o na variavel de resultado final
    }
    return(testes) #retorna o resultado dos testes
  }
  else # se o numero de colunas das tabelas for diferente o programa
```

```
para e vai pra mensagem abaixo
{
    stop("número de colunas das tabelas não é igual") #comando de
parada, e retorno de mensagem
}
}
```

funcao do dia 12 de abril, consegui fazer graficos, teste de normalidade, e teste não parametrico no lugar da simulacao que pensei inicialmente

[help2.txt](#)

```
test.tabelas=function(tab,tab1) #insira aqui suas tabelas
{
  if(ncol(tab)==ncol(tab1)) #se tabelas tiverem mesmo numero de colunas,
  siga
  {
    dim.tab=dim(tab)[2]      #numero de colunas da tabela 1 em um vetor
    dim.tabx=dim(tab1)[2]   #numero de colunas da tabela 2 em um vetor
    testes=c(0)             #saida para os dados
    for (i in 1:(dim.tab))  #contador de 1 até numero de colunas das tabelas
    {
      if(ks.test(tab[,i],"pnorm")$p.value>0.05 &
      ks.test(tab1[,i],"pnorm")$p.value>0.05) #se teste de normalidade para as
      duas colunas der positivo, seguir para testeT
      {
        x11()                #abre nova janela para graficos
        par(mfrow = c(2,3))  #janela tem duas linhas e 3 colunas
        plot(density(tab[ ,i])) #grafico de densidade para tabela 1
        boxplot(tab[ ,i], xlab= i) #boxplot para tabela 1
        qqnorm(tab[ ,i])      #qqnorm para tabela 1
        qqline(tab[ ,i])      #qqline no qqnorm

        plot(density(tab1[ ,i])) #grafico de densidade para tabela 2
        boxplot(tab1[ ,i], xlab= i) #boxplot para tabela 2
        qqnorm(tab1[ ,i])      #qqnorm para tabela 2
        qqline(tab1[ ,i])      #qqline no qqnorm

        testes[i]= t.test(tab[,i],tab1[,i])[3] #teste t comparando medias
        das duas colunas
      }
      else{ #se teste de normalidade para uma das duas colunas der negativo
      <0,05 , seguir para teste nao parametrico de wilcox
        x11()                #abre nova janela para graficos
        par(mfrow = c(2,3))  #janela tem duas linhas e 3 colunas
        plot(density(tab[ ,i])) #grafico de densidade para tabela 1
        boxplot(tab[ ,i], xlab= i) #boxplot para tabela 1
        qqnorm(tab[ ,i])      #qqnorm para tabela 1
        qqline(tab[ ,i])      #qqline no qqnorm
```

```

    plot(density(tab1[,i]))    #grafico de densidade para tabela 2
    boxplot(tab1[,i], xlab= i)  #boxplot para tabela 2
    qqnorm(tab1[,i])          #qqnorm para tabela 2
    qqline(tab1[,i])          #qqline no qqnorm

    testes[i] = wilcox.test(tab[,i],tab1[,i],exact = F,correct =
T)[3] #teste de wilcox comparando medianas das duas colunas
  }
}
return(testes) #retorna o resultado dos testes
}
else
{
  stop("numero de colunas das tabelas n?o ? igual") #para o teste e
informa mensagem se o numero de colunas das duas tabelas for diferente
}
}

```

## Plano B

Serão feitas análises de comparação de correlações entre variáveis quantitativas.

Duas tabelas contem dados de variáveis quantitativas (1 ou mais colunas). As tabelas possuem as mesmas variáveis. Cada tabela representa um grupo. Serão feitas correlações 2 a 2, entre as variáveis de uma mesma tabela. Tais correlações serão feitas para as duas tabelas. Por fim serão comparadas as correlações obtidas para cada grupo por um teste de comparação de correlações.

ex. correlação da coluna 1 versus 2 na tabela 1 será comparada com a correlação da coluna 1 versus 2 na tabela 2.

From:

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

Permanent link:

[http://ecor.ib.usp.br/doku.php?id=05\\_curso\\_antigo:r2015:alunos:trabalho\\_final:andre.vaquero:start](http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2015:alunos:trabalho_final:andre.vaquero:start)

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