



A permutation test for the race model inequality

Gondan, Matthias

Published in:
Behavior Research Methods

DOI:
[10.3758/BRM.42.1.23](https://doi.org/10.3758/BRM.42.1.23)

Publication date:
2010

Document version
Publisher's PDF, also known as Version of record

Citation for published version (APA):
Gondan, M. (2010). A permutation test for the race model inequality. *Behavior Research Methods*, 42(1), 23-28.
<https://doi.org/10.3758/BRM.42.1.23>

Supplementary Material

A script in R statistical programming language (R development core team, 2009) and a Matlab script (R2007b, The MathWorks, Munich, Germany), as well as usage instructions and example data from a redundant target experiment (Gondan et al., 2004) are provided as supplementary material for *A permutation test for the race model inequality*:

- `rmiperm.r`: R script for the permutation test
- `rmiperm.m`: Matlab script
- `go2004.dat`: ASCII file with example data from Gondan et al. (2004)

In the scripts, it is assumed that these files have been downloaded to the local folder `C:\RTDATA`. The current version of R for Windows and other operating systems can be downloaded from <http://www.r-project.org/>, in general, Windows users download the binary package of the program. After installation of the R program, “Open Script...” can be chosen from the “File” menu to open `rmiperm.r` which contains the code for the permutation test. The entire script is run by choosing “Execute all” from the “Edit” menu, thereby analyzing the example data (`C:\RTDATA\go2004.dat`). This path can be modified to point to the correct results file. The results file should be in tab-delimited ASCII text format and should include the three following columns:

- `obs`: observer number
- `cond`: condition (A, V, AV)
- `rt`: response time in Milliseconds, with omitted responses coded as “Inf”

The three column names should be included as the header of the results file, thus, the first few lines of the results file look like:

```
obs cond  rt
  1   A  245
  1   V  281
  1  AV  212
  1   V  Inf
```

If default options are used, the command `rmi_perm` performs a permutation test using the quantiles .05, .10, .15, .20, .25, .30 of the entire response time distribution. By default, 10001 permutations are used to generate the distribution of the t_{\max} statistic under the race model assumption. These default options can be adjusted using the following arguments:

- `q`: quantiles to be used for the test, default: .05, .10, .15, .20, .25, .30
- `nperm`: number of permutations used for determining the distribution of t_{\max} under the race model. Use an odd number like 1001 or 10001 (default) to avoid P values corresponding exactly to 0.05.

The return value is a list with two elements, the observed t_{\max} statistic and one-sided P value. Significant P values indicate a violation of the race model. Functions, arguments, defaults and return value of the Matlab script (`rmi_perm.m`) are identical to those of the R script.

R script: rmiperm.r

```
#
# Permutation test for the race model inequality
#
# Tested with R-2.9.1, download at http://www.stats.bris.ac.uk/R/
#
# tmax statistic
tmax = function(di)
{
  # t-Tests for each percentile
  tvalues = rowMeans(di) / apply(di, 1, sd) * sqrt(ncol(di))
  max(tvalues)
} # tmax

# The significance test
#
# - d: list including data of all participants
# - quantiles: quantiles of AV distribution at which the RMI is to be tested
# - nperm: number of random permutations used to generate H0 distribution
# - return value: tmax, critical value for one-sided test, P value
rmi_perm = function(d, quantiles=c(.05, .10, .15, .20, .25, .30), nperm=10001)
{
  # Determine quantiles from entire response time distribution
  tmix      = lapply(d, unlist)
  q         = lapply(tmix, quantile, probs=quantiles, type=4)

  # Evaluate distribution functions at q
  FA        = mapply(function(ti, qi) ecdf(ti$A)(qi), d, q)
  FV        = mapply(function(ti, qi) ecdf(ti$V)(qi), d, q)
  FAV       = mapply(function(ti, qi) ecdf(ti$AV)(qi), d, q)

  # Determine di = FAV(q) - FA(q) - FV(q)
  di        = matrix(FAV - FA - FV, nrow=length(quantiles), ncol=length(d))

  # Observed test statistic
  stat      = tmax(di)

  # Permutation distribution of test statistic
  stati = numeric(nperm)
  for(i in 1:nperm)
  {
    stati[i] = tmax(di %*% diag(sign(rnorm(d))))
  }

  list(tmax=stat, tcrit=quantile(stati, 0.95), P=mean(stat <= stati))
} # rmi_perm

# Read in data from text file
#
# obs: observer number, cond: condition, rt: response time (Inf: omitted response)
#
# obs cond  rt # This header should be included in the file
# 1   A   245 # response to A
# 1   V   281 # response to V
# 1  AV   212 # response to AV
# 1   V  Inf # omitted response
rmi_read = function(fname)
{
  d = read.table(fname, header=TRUE)
  obs = split(d[, c('cond', 'rt')], d$obs)
  for(i in 1:length(obs))
    obs[[i]] = split(obs[[i]]$rt, obs[[i]]$cond)
  obs
} # rmi_read

# Example session with data in C:\RTDATA\go2004.dat
data = rmi_read('C:/RTDATA/go2004.dat')
rmi_perm(data, quantiles=c(.05, .10, .15, .20, .25, .30), nperm=10001)
```

```

function [p, tmax, tcrit] = rmiperm(fname, onlyobs, q, nperm)
% Permutation test of the race model inequality (Tested with Matlab version 7.0.1)
%
% fname: filename to be read
%   obs: Observer number
%   cond: Condition (A, V, AV)
%   rt:  response time
%   Inf: Omitted response
%
%   obs cond   rt
%   1   A   234
%   1   AV  263
%   1   V   298
%   1   A   Inf
%
% onlyobs (optional): restrict analysis to these observers in the analysis
%   default []: use all observers
%
% q: quantiles of overall response time distribution at which RMI should be tested
%   default 0.05:0.05:0.30
%
% nperm: number of permutations used to determine tmax distribution, default 10001
%
% defaults if parameters are omitted
if nargin < 1
    fname = 'c:\rtdata\go2004.dat'
end

if nargin < 2
    onlyobs = []
end

if nargin < 3
    q = 0.05:0.05:0.30
end

if nargin < 4
    nperm = 10001
end

% default return values
p = -1 ;
tcrit = -1 ;
tmax = 'error' ;

% read file
[fid, message] = fopen(fname, 'r') ;
if fid == -1
    disp(message)
    return
end

header = textscan(fid, '%s %s %s', 1) ;
if header{1}{1} ~= 'obs'
    disp('Error reading header.')
    return
end

if header{2}{1} ~= 'cond'
    disp('Error reading header.')
    return
end

if header{3}{1} ~= 'rt'
    disp('Error reading header.')
    return
end

data = textscan(fid, '%f %s %f', 'headerlines', 1) ;
fclose(fid) ;

```

```

% column 1: observer no
% column 2: condition (A, V, AV)
% column 3: response time
obs = data{1} ;
cond = data{2} ;
rt = data{3} ;

% Extract response times
if isempty(onlyobs)
    onlyobs = min(obs):max(obs) ;
end

d = [] ;
for i = onlyobs
    tA = rt(obs == i & strcmp('A', cond)) ;
    tV = rt(obs == i & strcmp('V', cond)) ;
    tAV = rt(obs == i & strcmp('AV', cond)) ;
    tAll = sort(rt(obs == i)) ;

    % Quantiles of overall RTs
    edges = tAll(round(q*length(tAll))) ;

    % Empirical CDFs
    maxrt = 1000 ; % adjust if necessary
    fA = histc(tA, 1:maxrt) ;
    fV = histc(tV, 1:maxrt) ;
    fAV = histc(tAV, 1:maxrt) ;

    cdfA = cumsum(fA)/length(tA) ;
    cdfV = cumsum(fV)/length(tV) ;
    cdfAV = cumsum(fAV)/length(tAV) ;

    % Race Model Difference
    d = [d ; cdfAV(edges)' - cdfA(edges)' - cdfV(edges)'] ;
end

% Observed tmax statistic
tmax = max(mean(d) ./ std(d)) * sqrt(length(onlyobs)) ;

% Distribution of tmax
tmaxi = zeros(1, nperm) ;
for i = 1:nperm
    dp = diag(sign(rand(1, length(onlyobs)) - 0.5)) * d ;
    tmaxi(i) = max(mean(dp) ./ std(dp)) * sqrt(length(onlyobs)) ;
end

% 95th quantile of simulated tmax
tmaxi = sort(tmaxi) ;
tcrit = tmaxi(round(0.95*nperm)) ;

% P value
p = sum(tmax < tmaxi)/nperm ;

if nargin < 3
    disp(sprintf('tmax=%.3f tcrit=%.3f P=%.3f', tmax, tcrit, p))
end

return

```