Tutorial how to run exact np tests with R

Karl H. Schlag[*]

September 15, 2015

Abstract

this document shows how to use nonparametric tests of Schlag (2008) using R, from installing R all the way to example data sets. any and all feedback most welcome.

1 Downloading R

i suggest to use Rstudio, download at https://www.rstudio.com/

2 Installing npExact package

go to https://github.com/zauster/npExact and click the "download zip" button and save the file (npExact-master.zip) into a folder say "test", then extract "npExact-master" the file into that folder

open Rstudio, on the lower right window click "install packages"
choose "install from" the option "package archive file (zip,tar.gz)"
then choose "browse" and go to the folder "test" and then to the subdirectory "npExact-master"
to find the file "npExact_0.2.tar.gz" and click on this and it will install

if you want to use the npExact package, tick the box in the lower right window in the sheet "packages".
to see an overview of the package, type:
  ? npExact
to get a help file for a specific program, like npMeanSingle, type
  ? npMeanSingle

3 Newer Tests on my homepage

download the file "np additional tests.r" from http://homepage.univie.ac.at/karl.schlag/ under research statistics
open the *.r file Rstudio, highlight all the text using contrl A, click "run" in upper right corner of window

[*]Department of Economics, University of Vienna, karl.schlag@univie.ac.at

4 Preparing Data

note that you can avoid specifying the path each time if you change the working directory:

```
setwd("C:/enter your path")
```

4.1 via Notepad

(simplest option but messy as each variable gets its own file so many files created)
i) Copy a single data vector as a column without header into a notepad document
ii) save the file as filename.dat with "All Files" type option (remember where you save it).
iii) In the "console" window in R write

```
X <- as.matrix(read.table("c:/enter your path/filename.dat"))
```

now X contains your data (eg check the mean by typing mean(X))

4.2 via Excel

(best option unless already in stata format, all variables go in one file)
save each variable in a columns with headers as a csv type file in excel
in the "console" window of R type:

```
M <- as.matrix(read.table("C:/enter your path/filename.csv",header=TRUE,sep=";"))
```

now assuming that your variable of interest is in column j, type

```
X <- M[,j]
```

4.3 via Stata

using data that comes from stata format type:

```
library(foreign)
M <- read.dta("C:/enter your path/filename.dta")
```

now assuming that your variable of interest is in column j, type

```
X <- M[,j]
```

5 Data Examples

5.1 two independent samples

index of minority share holder protection on scale [0,1]:
German<-c(0.18, 0.21, 0.25, 0.27, 0.28, 0.29, 0.29, 0.32, 0.33, 0.47, 0.5, 0.56, 0.65, 0.76)
Common<-c(0.21, 0.35, 0.39, 0.39, 0.41, 0.41, 0.43, 0.58, 0.64, 0.65, 0.67, 0.73, 0.76, 0.79, 0.81, 0.81, 0.95, 0.95, 0.95, 0.96, 1)

5.2 matched pairs

level of pain before and after an operation on scale [0,100]
before <- c(80, 40, 50, 69, 50, 60, 80, 60, 80, 45, 84, 67, 50, 36, 68, 80, 68, 86, 78, 78, 82, 75, 86, 80, 77)

after <- c(78, 36, 45, 60, 41, 45, 64, 42, 61, 26, 64, 35, 16, 0, 20, 30, 17, 30, 22, 22, 24, 14, 22, 12, 8)

6 A Single Sample

6.1 Mean

(uses package npExact)
i) to test $H_0 : E\left(German\right) \leq 0.2$ at level 0.05 type
npMeanSingle(German,mu=0.2,low=0,up=1,alpha=0.05,alternative="greater")
so H_0 is rejected at $\alpha = 0.05$.
ii) to find p value look for lowest value of alpha st H_0 is just barely rejected: pvalue 0.002
iii) to find left hand boundary of 95% confidence interval of German find smallest value of mu such that not rejected at alpha=0.05: mu=0.25
iv) to test $H_0 : E\left(German\right) \geq 0.5$ at level 0.05 type
npMeanSingle(German,mu=0.5,low=0,up=1,alpha=0.05,alternative="less")

6.2 Variance

(uses package npExact)
to test $H_0 : Var\left(Common\right) \leq 0.01$ type
npVarianceSingle(Common,v=0.01, low=0,up=1,alpha=0.05,alternative="greater")
so H_0 is rejected at $\alpha = 0.05$.

7 Two Independent Samples (IS)

7.1 Comparing Means given IS

(uses package npExact)
to test $H_0 : E\left(Common\right) \leq E\left(German\right)$ at level $\alpha = 0.05$ type
npMeanUnpaired(German,Common,low=0,up=1,alpha=0.05,alternative="less")
so H_0 is not rejected at $\alpha = 0.05$, the p value is 0.07 (set "iterations=300000") to be sure that result is correct

7.2 Comparing Medians given IS

(uses "np Additional Tests Unpaired.r" from my homepage)
to test $H_0 : med(Common) = med(German)$ at level $\alpha = 0.1$, type
npMedianUnpaired(German,Common,alpha=0.1)
to obtain that it is rejected in favor of median(Common)>med(German)

7.3 Looking at the Stochastic Difference given IS

(uses package npExact)
to test $H_0 : P(Common > German) \leq P(Common < German)$ at level $\alpha = 0.05$, type
npStochinUnpaired(Common,German,alpha=0.05,alternative="greater")

to find a rejection in favor of $P(Common > German) > P(Common < German)$, so common index tending to be larger than the german index
the estimated stochastic difference $P(Common > German) - P(Common < German)$ is 0.66

7.4 Looking at Median Difference given IS

(uses "np Additional Tests Unpaired.r" from my homepage and as input routine npStochin from package npExact)
to test $H_0 : med(Common - German) = -0.1$ at level 0.01, type
    npMedianDiffUnpaired(Common,German,d=-0.1,alpha=0.01,estimate=T)
to find rejection in favor of median difference > -0.1
the estimated median difference between Common and German is equal to 0.284

8 Matched Pairs (MP)

8.1 Comparing Means given MP

(uses package npExact)
to test $H_0 : E(before) \leq E(after)$ at level $\alpha = 0.05$ type
    npMeanPaired(before,after,low=0,up=100,alpha=0.05,alternative="greater")
and find that it is rejected in favor of $E(before) > E(after)$, p value is 0.02

8.2 Comparing Medians given MP

(uses "np Additional Tests Paired.r" from my homepage)
to test $H_0 : med(before) = med(after)$ at level $\alpha = 0.1$, type
    npMedianPaired(before,after,alpha=0.1)
to obtain that it is rejected in favor of med(before)>median(after)
the estimated difference in medians equal to 45

8.3 Looking at the Stochastic Difference given MP

(uses "np Additional Tests Paired.r" from my homepage)
to test $H_0 : P(before > after) = P(before < after)$ at level $\alpha = 0.05$, type
    npStochinPaired(before,after,alpha=0.05)
to find a rejection in favor of pain before tending to be larger than pain after
the estimated stochastic difference $P(before > after) - P(before < after)$ is 1

8.4 Looking at Median Difference given MP

(uses "np Additional Tests Paired.r" from my homepage)
to test $H_0 : med(before - after) = 0$ at level 0.01, type
    npMedianDiffPaired(before, after,d=0,alpha=0.01)
to find a rejection in favor of the median change in pain due to the operation being positive
the estimated median change in pain being equal to 34