Part VIII

TESTS ON RANDOMNESS

To test if a sequence of numbers, occurrences of diseased cases, or other data are randomly drawn from an underlying distribution or not is the topic of this part. Even though this might not sound difficult, it is. Let us assume a coin is tossed 10 times with five times heads and five times tails. This is a result we would expect, at least in the long run. However, if the sequence of (H)eads and (T)ails is `HHHHHTTTTT` you would argue against the hypothesis that the sample is random. The sequence `HTHTHTHTHT` looks 'more' random, but a little bit artificial. A sequence such as `HHTHTHTTTH` seems to be 'more' random than the two sequences before. So the question remains how to decide on randomness. Computer programs can only produce pseudo-random numbers due to algorithms generating such numbers. Nevertheless, if the cycle is long enough these numbers should appear to be random. A test on randomness can be used to detect if a random number generator works well (do not reject the hypothesis of randomness) or not (reject the hypothesis of randomness).

13

Tests on randomness

Section 13.1 deals with *run tests*. The *Wald–Wolfowitz runs test*, also known as the *single-sample runs test*, and the *runs up and down test* are the most common ones. For the definition of a *run* please refer to the Glossary (Appendix C). Section 13.2 covers successive difference tests (tests based on the difference of consecutive observations). We introduce the *von Neuman test* and its rank version also known as *Bartels' test*. All these tests are useful to test if a dataset is sampled randomly from the same distribution.

13.1 Run tests

13.1.1 Wald–Wolfowitz runs test

Description: Tests if a sample is sampled randomly from an underlying population.

Assumptions:
- Data are at least measured on a nominal scale.
- Let $X_1, \ldots, X_n$ be a sequence of binary random variables with observations $x_1, \ldots, x_n$. These may also be derived as dichotomized versions of random variables with originally multinomial or quantitative output.
- The n observations thereby contain n_1 elements with one of the outputs and $n - n_1 = n_2$ with the other one.

Hypotheses:
(A) H_0 : Sequence $X_1, \ldots, X_n$ is randomly generated
vs H_1 : Sequence is not randomly generated
(B) H_0 : Sequence $X_1, \ldots, X_n$ is randomly generated
vs H_1 : Sequence has tendency to mix
(C) H_0 : Sequence $X_1, \ldots, X_n$ is randomly generated
vs H_1 : Sequence has tendency to cluster

Test statistic: $Z = \left(R - 1 - \frac{2n_1n_2}{n}\right)\sqrt{\frac{2n_1n_2(2n_1n_2-n)}{n^2(n-1)}}$, where R is the number of runs in the sequence.

Test decision: Reject H_0 if for the observed value z of Z
(A) $z < z_{\alpha/2}$ or $z > z_{1-\alpha/2}$
(B) $z > z_{1-\alpha}$
(C) $z < z_{\alpha}$

p-values: (A) $p = 2\Phi(-|z|)$
(B) $p = 1 - \Phi(z)$
(C) $p = \Phi(z)$

Annotations:

- For the number of runs R see the Glossary (Appendix C) for an explanation.
- z_α is the α-quantile of the standard normal distribution.
- The number of runs R is approximately normally distributed with mean $(2n_1n_2)/n + 1$ and variance $(2n_1n_2(2n_1n_2 - n))/(n^2(n-1))$.
- The test statistic Z follows a standard normal distribution.
- The normal approximation follows from Wald and Wolfowitz (1940), where the authors focus on testing whether two samples are drawn from the same population. Nevertheless, the test is sometimes called the Wald–Wolfowitz runs test. It is also known as the *single-sample runs test*.
- Critical values for small samples ($n_1, n_2 \leq 20$) are given in Swed and Eisenhart (1943).
- In test problem (B) it is tested against a tendency to mix, which means that the sequence shows too many runs to be generated randomly. For example, we may have a dichotomized random variable with realizations A and B in following sequence: $\underline{\text{A}}\ \underline{\text{B}}\ \underline{\text{A}}\ \underline{\text{B}}\ \underline{\text{A}}\ \underline{\text{B}}\ \underline{\text{A}}\ \underline{\text{B}}$. In problem (C) the alternative states a tendency to cluster, hence too few runs, for example, in the extreme case that $\underline{\text{AAAA}}\ \underline{\text{BBBB}}$.
- A continuity correction may be applied to the test statistic (Gibbons 1988).

Example: To test the hypothesis that a specific coin is fair. A sequence of 15 coin tosses is observed (dataset in Table A.7).

SAS code

```
* Go through the dataset and count the head and tail runs,
* the final count contains all relevant information;
data runs;
 set coin;
 retain headruns 0 tailruns 0 n1 0 n2 0;
 temp1=head;       * Let temp1 be the same as head ;
 temp2=lag(head);  * temp2 is a lagged variable of head
                     (a shift of one place);

 * Count the number of heads (n1) and tails (n2);
 if head=1 then n1=n1+1;
 if head=0 then n2=n2+1;

 * Look at the first observation to decide if
 * a head run or a tail run starts;
 if _N_=1 then do;
   if head=1 then headruns=1;
   if head=0 then tailruns=1;
 end;

 * Go through the dataset for all other observations
 * if the value is not the shifted value then a new
     run starts;
 if _N_>1 then do;
  if temp1 ^= temp2 then do;
   if temp1=1 then headruns=headruns+1;
   if temp1=0 then tailruns=tailruns+1 ;
  end;

end;

run;

* Calculate test statistic and p-values;
data wwtest;
 set runs nobs=nobs;
 drop temp1 temp2;
 format p_value_A p_value_B p_value_C pvalue.;

 if _n_=nobs; * Keep last observation of the dataset 'runs';

 r=(headruns+tailruns); * Total number of runs;
 n=n1+n2;               * Number of observations;

 * Calculate test statistic;
 mu=2*(n1*n2)/n+1;
 sigma=sqrt((2*n1*n2)*(2*n1*n2-n)/(n**2*(n-1)));
 z=(r-mu)/sigma;

 p_value_A=2*probnorm(-abs(z));
 p_value_B=1-probnorm(z);
 p_value_C=probnorm(z);

run;

* Output results;
proc print split='*' noobs;
```

```
 var z r p_value_A p_value_B p_value_C;
 label z='Test statistic Z*----------------'
       r='No. of Runs*-----------'
       p_value_A='p-value A*----------'
    p_value_B='p-value B*----------'
    p_value_C='p-value C*----------';
 title 'Wald-Wolfowitz Test';
run;
```

SAS output

```
Wald-Wolfowitz Test

Test statistic Z  No. of Runs  p-value A   p-value B
----------------  -----------  ----------  ----------
0.82565                10        0.4090      0.2045

p-value C
----------
  0.7955
```

Remarks:

- The above code uses the *retain* statement. This function lets the variable retain its value from one observation to the next. The 0 after the variable name initializes the variable with a value of zero. The *lag* function shifts the value so that observation 2 gets the value of observation 1 and observation 3 gets the value of observation 2 and so on. This is useful to determine if a run goes on or is interrupted.
- The SAS Institute gives an example code of this test on their website (http://support.sas.com/kb/33/092.html).

R code

```
# Set the number of head and tail runs to zero
headruns=0
tailruns=0

# Get the number of observations and the number
# of heads and tails
n<-length(coin$head)
n1<-length(coin$head[coin$head==1])
n2<-length(coin$head[coin$head==0])

# Look at the first observation to decide if
# a head run or a tail run starts

if (coin$head[1]==1) headruns<-1
if (coin$head[1]==0) tailruns<-1
```

```
# Go through the dataset for all other observations
# if the value is not the value of the former
# observation a new run starts
for (i in 2:n) {
 if (coin$head[i] != coin$head[i-1]) {
  if (coin$head[i]==1) headruns<-headruns+1
  if (coin$head[i]==0) tailruns<-tailruns+1
 }
}

# Calculate test statistic and p-values
r<-headruns+tailruns
mu<-2*n1*n2/n+1
sigma<-sqrt((mu-1)*(mu-2)/(n-1))

z<-(r-mu)/sigma;

p_value_A<-2*pnorm(-abs(z))
p_value_B<-1-pnorm(z)
p_value_C<-pnorm(z)

# Output results
cat("Wald-Wolfowitz Runs Test",
"\n------------------------","\n")
z
r
p_value_A
p_value_B
p_value_C
```

R output

```
Wald-Wolfowitz Runs Test
------------------------
> z
[1] 0.8256519
> r
[1] 10
> p_value_A
[1] 0.4090016
> p_value_B
[1] 0.2045008
> p_value_C
[1] 0.7954992
```

Remarks:

- Here, we used the even simpler term $(mu - 1) * (mu - 2)/(n - 1)$ for the variance of the approximated normal distribution of the runs R. This term is identical to the one given in the above annotations.

13.1.2 Runs up and down test

Description: Tests if a sample is sampled randomly from an underlying population.

Assumptions:

- Data are at least measured on an ordinal scale.
- Let $X_1, \dots, X_n$ be a sequence of random variables with observations $x_1, \dots, x_n$.
- The sequence is transformed into a sequence of length $n-1$ consisting solely of $+$ and $-$ signs. The signs are determined by taking the sign of the successive differences $X_{i+1} - X_i$, $i = 1, \dots, n-1$ (Gibbons 1988).

Hypotheses:

(A) H_0 : Sequence $X_1, \dots, X_n$ is randomly generated
vs H_1 : Sequence is not randomly generated
(B) H_0 : Sequence $X_1, \dots, X_n$ is randomly generated
vs H_1 : Sequence tends to show rapid oscillation
(C) H_0 : Sequence $X_1, \dots, X_n$ is randomly generated
vs H_1 : Sequence tends to have a trend or gradual oscillation

Test statistic: $Z = (V - (2n-1)/3)\sqrt{(16n-29)/90}$, where V equals the number of runs in the transformed sequence with $+$ and $-$ signs.

Test decision: Reject H_0 if for the observed value z of Z
(A) $z < z_{\alpha/2}$ or $z > z_{1-\alpha/2}$
(B) $z > z_{1-\alpha}$
(C) $z < z_\alpha$

p-values:
(A) $p = 2\Phi(-|z|)$
(B) $p = 1 - \Phi(z)$
(C) $p = \Phi(z)$

Annotations:

- This test was introduced by Wallis and Moore (1941) for quantitative data observed in a time series. They use the number of complete runs $V - 2$ as the test statistic. It (and thereby also V) is asymptotically normal with $E(V-2) = (2n-7)/3$, such that $E(V) = (2n-1)/3$, and variance $Var(V-2) = Var(V) = (16n-29)/90$.
- The test statistic Z asymptotically follows a standard normal distribution.
- Critical values for this test for sample sizes less than 25 can be found in Edgington (1961).
- A continuity correction may be applied to the test statistic (Wallis and Moore 1941).
- If the difference $X_{i+1} - X_i$ is zero it is not clear if it belongs to an up or down run. Hence, this might change the number of runs; we refer to Wallis and Moore (1941) on how to deal with this situation.

Example: To test the hypothesis that the yearly harvest of wheat (in million tons) in Hyboria is randomly generated. The dataset contains a sequence of 10 observations from 2002 to 2011 (dataset in Table A.8).

SAS code

```
* Convert the values in a sequence of "+" for a run up
* and a "-" for a run down;
data runs1;
  set harvest;
   d=dif(harvest);
   if dif(harvest)<0 then vec="-";
   if dif(harvest)>0 then vec="+";
run;

* Get rid of first observation;
data runs2;
 set runs1(firstobs=2);
run;

* Calculate runs;
data runs3;
 set runs2;
 retain upruns 0 downruns 0;

 temp1=vec;        * Let temp1 be the same as vec;
 temp2=lag(vec);   * temp2 is a lagged variable of vec
                     (a shift of one place);

 * Detect the starting run;
 if _N_=1 then do;
   if vec='+' then upruns=1;
   if vec='-' then downruns=1;
 end;

 * Go through the dataset for all other observations
 * and count runs up and runs down;
 if _N_>1 then do;
  if temp1 ^= temp2 then do;
   if temp1='+' then upruns=upruns+1;
   if temp1='-' then downruns=downruns+1 ;
  end;
 end;

run;

* Calculate test statistic and p-values;
data runs4;
 set runs3 nobs=nobs;
 drop temp1 temp2;
 format p_value_A pvalue.;

 if _n_=nobs; *Keep last observation of the dataset 'runs3';
 v=(upruns+downruns); * Total number of runs;
 n=_N_+1;             * Number of observations;

 * Calculate test statistic;
  mu=(2*n-1)/3;
  sigma=sqrt((16*n-29)/90);
  z=(v-mu)/sigma;

  p_value_A=2*probnorm(-abs(z));
  p_value_B=1-probnorm(z);
  p_value_C=probnorm(z);
run;

* Output results;
proc print split='*' noobs;
 var z v p_value_A p_value_B p_value_C;
 label z='Test statistic Z*----------------'
       v='No. of Runs*-----------'
       p_value_A='p-value A*----------'
    p_value_B='p-value B*----------'
    p_value_C='p-value C*----------';
 title 'Runs Up and Down Test';
run;
```

SAS output

```
                 Runs Up and Down Test

Test statistic Z    No. of Runs    p-value A
----------------    -----------    ----------
    0.55258              7           0.5806

p-value B    p-value C
----------  ----------
 0.29028     0.70972
```

Remarks:

- The *dif* function calculates the difference between the value of an observation and the value of the prior observation.
- The above code uses the *retain* statement. This function lets the variable retain its value from one observation to the next. The 0 after the variable name initializes the variable with a value of zero.

R code

```
# Store the harvest figures in the variable x
x<-harvest$harvest

# Get number of observations
```

```
n<-length(x)

# Define some variables
upruns<-0       # holds the number of runs up
downruns<-0     # holds the numer of down runs

# Define a vector to store different characters for an
# run up "+" and run down "-"
vec<-vector("character", length = n-1)

# Go through the observations and assign "+" for an
# run up and a "-" for a run down
for (i in 2:n){
if (x[i] < x[i-1])  vec[i-1]="-"
if (x[i] > x[i-1])  vec[i-1]="+"
}

# Detect the starting run
if (vec[1]=="+") upruns<-1
if (vec[1]=="-") downruns<-1

# Go through the runs vector and count
# runs up and runs down
for (j in 2:length(vec)){
 if (vec[j] != vec[j-1]){
  if (vec[j-1]=="+") upruns<-upruns+1
  if (vec[j-1]=="-") downruns<-downruns+1
 }
}

# Calculate test statistic and p-values
v<-upruns+downruns
mu<-(2*n-1)/3
sigma<-sqrt((16*n-29)/90)
z<-(v-mu)/sigma
p_value_A<-2*pnorm(-abs(z))
p_value_B<-1-pnorm(z);
p_value_C<-pnorm(z);

# Output results
cat("Runs Up and Down Test \n\n",
"V       ","Z      ","p-value A"," ",
"p-value B"," ","p-value C","\n",
"---  --------- ----------- ----------- -----------","\n",
v,"  ",z," ",p_value_A," ",p_value_B," ",p_value_C,"\n")
```

R output

```
Runs Up and Down Test

 V        Z      p-value A   p-value B   p-value C
 ---  --------- ----------- ----------- -----------
 7    0.552579   0.5805517   0.2902759   0.7097241
```

Remarks:

- There is no basic R function to calculate this test directly.

13.2 Successive difference tests

13.2.1 von Neumann test

Description: Tests if a sample is sampled randomly from an underlying normal population.

Assumptions:
- Data are at least measured on an ordinal scale.
- Let $X_1, \dots, X_n$ be a sequence of random variables with observations $x_1, \dots, x_n$.
- $X_1, \dots, X_n$ follow the same normal distribution.

Hypotheses:
(A) H_0 : Sequence $X_1, \dots, X_n$ is randomly generated
vs H_1 : Sequence is not randomly generated
(B) H_0 : Sequence $X_1, \dots, X_n$ is randomly generated
vs H_1 : Sequence tends to short oscillation
(C) H_0 : Sequence $X_1, \dots, X_n$ is randomly generated
vs H_1 : Sequence tends to have a trend or gradual oscillation

Test statistic: $Z = \left(1 - \frac{V}{2}\right)\sqrt{(N-2)/(N^2-1)}$,
where V is the so-called von Neumann ratio:

$$V = \sum_{i=1}^{N-1} (X_{i+1} - X_i)^2 \Bigg/ \sum_{i=1}^{N} (X_i - \overline{X})^2$$

Test decision: Reject H_0 if for the observed value z of Z
(A) $z < z_{\alpha/2}$ or $z > z_{1-\alpha/2}$
(B) $z > z_{1-\alpha}$
(C) $z < z_\alpha$

p-values:
(A) $p = 2\Phi(-|z|)$
(B) $p = 1 - \Phi(z)$
(C) $p = \Phi(z)$

Annotations:
- The test is based on the ratio of the sum of squared differences of consecutive observations to the empirical variance. Small and large differences point to non-randomness (von Neumann 1941; Gibbons 1988).

- Young (1941) showed that $\left(1 - \frac{V}{2}\right)$ is asymptotically normally distributed with mean zero and variance $(N-2)/(N^2-1)$ for normal samples.
- The test statistic Z follows a standard normal distribution.
- Critical values for this test for sample sizes between 4 and 60 can be found in Young (1941) and Hart (1942).

Example: To test the hypothesis that the yearly harvest of wheat (in million tons) in Hyboria is randomly generated. The dataset contains a sequence of 10 observations from 2002 to 2011 (dataset in Table A.8).

SAS code

```
* Calculate the square of the differences;
data numerator;
 set harvest;
 d=dif(harvest)**2;
run;

* Calculate the sum of the squared differences;
proc means data=numerator sum;
 var d;
 output out=numerator2 sum=numerator;
run;

* Calculate the empirical variance of the values;
proc means data=harvest var;
 var harvest;
 output out=denominator var=variance;
run;

* Calculate test statistic and p-values;
data neumann;
 merge numerator2 denominator;

 format p_value_A p_value_B p_value_C pvalue8.;

 n=_FREQ_;
 vn=numerator/((n-1)*variance);
 z=(1-(vn/2))/sqrt((n-2)/(n**2-1));

 p_value_A=2*probnorm(-abs(z));
 p_value_B=1-probnorm(z);
 p_value_C=probnorm(z);
run;

* Output results;
proc print split='*' noobs;
 var vn z p_value_A p_value_B p_value_C ;
 label vn='von Neumann statistic*--------------------'
       z='Test statistic Z*-----------'
       p_value_A='p-value A*----------'
    p_value_B='p-value B*----------'
    p_value_C='p-value C*----------';
 title 'von Neumann Test';
run;
```

SAS output

```
                 von Neumann Test

von Neumann statistic    Test statistic Z    p-value A
--------------------    ----------------    ----------
      2.88641              -1.55911         0.118971

 p-value B    p-value C
----------   ----------
 0.94051      0.059486
```

Remarks:

- The above code uses the *dif* function. This function calculates the successive differences of the values of a variable.
- We use *proc means* to calculate the empirical variance and must therefore multiply the result with $n - 1$ to calculate the von Neumann ratio.

R code

```
# Store the harvest figures in the variable x
x<-harvest$harvest

# Get number of observations
n<-length(x)

# The next vector holds the n-1 differences
numerator<-seq(0,0,length=n-1)

# Go through the observations and calculate
# the square of the differences
for (i in 1:n-1){
 numerator[i]<-(x[i+1]-x[i])^2
}

# Calculate the von Neumann test statistic
m<-mean(x)
VN<-sum(numerator)/sum((x-m)^2)

# Calculate the normal approximation
z=(1-(VN/2))/sqrt((n-2)/(n^2-1))
```

```
# Calculate p-values
p_value_A<-2*pnorm(-abs(z));
p_value_B<-1-pnorm(z);
p_value_C<-pnorm(z);

# Output results
cat("von Neumann Test \n\n",
"von Neumann statistic ","    Z     ",
"p-value A","  ","p-value B","  ","p-value C","\n",
"---------------------  --------- ----------- ",
"----------- -----------","\n",
"     ",VN,"       ",z," ",p_value_A," ",
p_value_B," ",p_value_C,"\n")
```

R output

```
von Neumann Test

von Neumann statistic      Z      p-value A
---------------------  --------- -----------
      2.886407         -1.559107   0.118971

 p-value B   p-value C
----------- -----------
 0.9405145   0.05948551
```

Remarks:

- A vector is used in R to store the squared differences.

13.2.2 von Neumann rank test (Bartels' test)

Description: Tests if a sample is sampled randomly from an underlying population.

Assumptions:
- Data are at least measured on an ordinal scale.
- Let $X_1, \dots, X_n$ be a sequence of random variables with observations $x_1, \dots, x_n$.

Hypotheses:
(A) H_0 : Sequence $X_1, \dots, X_n$ is randomly generated
vs H_1 : Sequence is not randomly generated
(B) H_0 : Sequence $X_1, \dots, X_n$ is randomly generated
vs H_1 : Sequence tends to short oscillation
(C) H_0 : Sequence $X_1, \dots, X_n$ is randomly generated
vs H_1 : Sequence tends to have a trend or gradual oscillation

Test statistic: $Z = (V - 2)\sqrt{(4/N)}$,
where V is the von Neumann's ratio of the ranks of the sample:
$$V = \sum_{i=1}^{N-1} (R_{i+1} - R_i)^2 \Big/ \sum_{i=1}^{N} (R_i - \overline{R})^2$$
with $R_i = rank(X_i)$ for $i = 1, \dots, n$

Test decision: Reject H_0 if for the observed value z of Z
(A) $z < z_{\alpha/2}$ or $z > z_{1-\alpha/2}$
(B) $z > z_{1-\alpha}$
(C) $z < z_\alpha$

p-values: (A) $p = 2\Phi(-|z|)$
(B) $p = 1 - \Phi(z)$
(C) $p = \Phi(z)$

Annotations:

- This is a rank version of van Neumann's test (Test 13.2.1) introduced by Bartels (1982). The test is also known as *Bartels' test*.
- Bartels (1982) showed that V is asymptotically normally distributed with mean 2 and variance $4/N$, such that the test statistic Z follows a standard normal distribution.
- Exact critical values are also given in Bartels (1982). For these values only the numerator $\sum_{i=1}^{N-1} (R_{i+1} - R_i)^2$ of the von Neumann ratio of the ranks is used.

Example: To test the hypothesis that the yearly harvest of wheat (in million tons) in Hyboria is randomly generated. The dataset contains a sequence of 10 observations from 2002 to 2011 (dataset in Table A.8).

SAS code

```
* Calculate the ranks of the observations;
proc rank data=harvest out=rank_data;
 var harvest;
run;

* Calculate the square of the differences;
data numerator;
 set rank_data;
 d=dif(harvest)**2;
run;

* Calculate the sum of the squared differences;
proc means data=numerator sum;
 var d;
 output out=numerator2 sum=numerator;
run;
```

```
* Calculate the empirical variance of the values;
proc means data=rank_data var;
 var harvest;
 output out=denominator var=variance;
run;

* Calculate test statistic and p-values;
data neumann;
 merge numerator2 denominator;

 format p_value_A p_value_B p_value_C pvalue8.;

 n=_FREQ_;
 rvn=numerator/((n-1)*variance);
 z=(rvn-2)/sqrt(4/n);

 p_value_A=2*probnorm(-abs(z));
 p_value_B=1-probnorm(z);
 p_value_C=probnorm(z);
run;

* Output results;
proc print split='*' noobs;
 var rvn z p_value_A p_value_B p_value_C;
 label rvn=
      'von Neumann rank statistic*-------------------------'
       z='Test statistic Z*----------------'
       p_value_A='p-value A*----------'
    p_value_B='p-value B*----------'
    p_value_C='p-value C*----------';
 title 'von Neumann Rank Test';
run;
```

SAS output

```
von Neumann rank statistic  Test statistic Z  p-value A
-------------------------  ----------------  ----------
         2.88485                1.39907       0.161793

p-value B   p-value C
----------  ----------
0.080896    0.919104
```

Remarks:

- With *proc rank* the ranks are calculated.
- The above code uses the *dif* function. This function calculates the successive differences of the values of a variable.
- We use *proc means* to calculate the empirical variance and must therefore multiply the result with $n - 1$ to calculate the von Neumann ratio.

R code

```
# Store the harvest figures in the variable x
x<-harvest$harvest

# Get number of observations
n<-length(x)

# Calculate the ranks of the observations
r<-rank(x,ties.method="average")

# The next vector holds the n-1 differences
numerator<-seq(0,0,length=n-1)

# Go through the observations and calculate
# the squared differences
for (i in 1:n-1){
 numerator[i]<-(r[i+1]-r[i])^2
}

# Calculate the von Neumann rank test statistic
m<-mean(r)
RVN<-sum(numerator)/sum((r-m)^2)

# Calculate the normal approximation
z<-(RVN-2)/sqrt(4/n)

# Calculate p-Values
p_value_A<-2*pnorm(-abs(z));
p_value_B<-1-pnorm(z);
p_value_C<-pnorm(z);

# Output results
cat("von Neumann Rank Test \n\n",
"von Neumann rank statistic ","    Z      ",
"p-value A","  ","p-value B","  ","p-value C","\n",
"--------------------------  ---------  ----------- ",
"----------- -----------","\n",
"         ",RVN,"         ",z," ",p_value_A," ",
p_value_B," ",p_value_C,"\n")
```

R output

```
von Neumann Rank Test

von Neumann rank statistic      Z       p-value A
--------------------------  ---------  -----------
           2.884848          1.399068   0.1617925

 p-value B    p-value C
----------- -----------
 0.08089625  0.9191037
```

Remarks:

- The function *rank* is used to calculate the ranks.
- A vector is used to store the squared differences of the ranks.

References

Bartels R. 1982 The rank version of von Neumann's rank test for randomness. *Journal of the American Statistical Association* **77**, 40–46.

Edgington E.S. 1961 Probability table for number of runs of signs of first differences in ordered series. *Journal of the American Statistical Association* **56**, 156–159.

Gibbons J.D. 1988 Tests of randomness. In *Encyclopedia of Statistical Sciences*, Vol. 8, pp. 555–562. John Wiley & Sons, Ltd.

Hart B.I. 1942 Significance levels for the ratio of the mean square successive difference to the variance. *Annals of Mathematical Statistics* **13**, 445–447.

Swed F.S. and Eisenhart C. 1943 Tables for testing randomness of grouping in a sequence of alternatives. *Annals of Mathematical Statistics* **14**, 66–87.

von Neumann J. 1941 Distribution of the ratio of the mean square successive difference to the variance. *Annals of Mathematical Statistics* **12**, 367–395.

Wald A. and Wolfowitz J. 1940 On a test whether two samples are from the same population. *Annals of Mathematical Statistics* **11**, 147–162.

Wallis W.A. and Moore G.A. 1941 A significant test for time series analysis. *Journal of the American Statistical Association* **36**, 401–409.

Young L.C. 1941 On randomness in ordered sequences. *Annals of Mathematical Statistics* **12**, 293–300.