E83-10123

AgRISTARS

SM-L2-04234 CR-169726
JSC-17816

A Joint Program for Agriculture and Resources Inventory Surveys Through Aerospace Remote Sensing

Soil Moisture

February 1982

FORTRAN IMPLEMENTATION OF FRIEDMAN'S TEST FOR SEVERAL RELATED SAMPLES

(E83-10123) FORTRAN IMPLEMENTATION OF FRIEDMAN'S TEST FOR SEVERAL RELATED SAMPLES (Lockheed Engineering and Management) 25 p HC A02/MF A01 CSCL 02C
N83-15793
Unclas
G3/43 00123

S. A. Davidson

Lockheed Engineering and Management Services Company, Inc.
1830 NASA Road 1, Houston, Texas 77258

Lyndon B. Johnson Space Center
Houston, Texas 77058

ORIGINAL PAGE IS
OF POOR QUALITY

1. Report No. JSC-17816; SM-L2-04234	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Fortran Implementation of Friedman's Test for Several Related Samples		5. Report Date February 1982
		6. Performing Organization Code
7. Author(s) S. A. Davidson		8. Performing Organization Report No. LEMSCO-17502
		10. Work Unit No.
9. Performing Organization Name and Address Lockheed Engineering and Management Services Company, Inc. 1830 NASA Road 1 Houston, Texas 77058		11. Contract or Grant No. NAS 9-15800
		13. Type of Report and Period Covered Technical Report
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Lyndon B. Johnson Space Center Houston, Texas 77058		14. Sponsoring Agency Code

15. Supplementary Notes
The Agriculture and Resources Inventory Surveys Through Aerospace Remote Sensing is a joint program of the U.S. Department of Agriculture, the National Aeronautics and Space Administration, the National Oceanic and Atmospheric Administration (U.S. Department of Commerce), the Agency for International Development (U.S. Department of State), and the U.S. Department of the Interior.

16. Abstract

The FRIEDMAN program is a FORTRAN-coded implementation of Friedman's nonparametric test for several related samples with one observation per treatment/block combination, or as it is sometimes called, the two-way analysis of variance by ranks. A detailed description of the FRIEDMAN program is given in this document. A test data set and its results are also presented to aid the potential users of this program.

17. Key Words (Suggested by Author(s)) Friedman's test Friedman's T-statistic Two-way analysis of variance by rank	18. Distribution Statement

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 25	22. Price*

*For sale by the National Technical Information Service, Springfield, Virginia 22161

SM-L2-04234
JSC-17816

FORTRAN IMPLEMENTATION OF FRIEDMAN'S TEST FOR SEVERAL RELATED SAMPLES

Job Order 71-324

This report describes activities of the Soil Moisture project of the AgRISTARS program.

PREPARED BY

S. A. Davidson

APPROVED BY

Jess G. Carnes
J. G. Carnes, Supervisor
Radiation Characterization Section

T. C. Minter
T. C. Minter, Jr., Manager
Development and Evaluation Department

LOCKHEED ENGINEERING AND MANAGEMENT SERVICES COMPANY, INC.

Under Contract NAS 9-15800

For

Earth Resources Research Division
Space and Life Sciences Directorate
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS

February 1982

LEMSCO-17502

PREFACE

The Agriculture and Resources Inventory Surveys Through Aerospace Remote Sensing is a multiyear program of research, development, evaluation, and application of aerospace remote sensing for agricultural resources, which began in fiscal year 1980. This program is a cooperative effort of the U.S. Department of Agriculture, the National Administration (U.S. Department of Commerce), the Agency for International Development (U.S. Department of State), and the U.S. Department of the Interior.

The work which is the subject of this document was performed by the Earth Resources Applications Division, Space and Life Sciences Directorate, Lyndon B. Johnson Space Center, National Aeronautics and Space Administration and Lockheed Engineering and Management Services Company, Inc. The tasks performed by Lockheed Engineering and Management Services Company, Inc., were accomplished under Contract NAS 9-15800.

CONTENTS

1. INTRODUCTION

The FRIEDMAN program is a FORTRAN-coded implementation of Friedman's nonparametric test for several related samples with one observation per treatment/block combination, or as it is sometimes called, the two-way analysis of variance by ranks. This test is particularly useful when the assumptions of normal theory fail to hold for a randomized, complete block design. In fact, for this test, the only assumptions made are that the blocks are mutually independent and that within each block the data may be ranked from smallest to largest. A test statistic is then formed based upon the ranks, not the actual data. This statistic is then used to test the null hypothesis, as follows:

H_0: Treatments have identical effects.

H_1: At least one treatment tends to yield larger observations than at least one other treatment.

For further details, see reference 1.

A detailed description of the FRIEDMAN program is given in this document. A test data set and its results are also presented to aid the potential users of this program.

2. PROGRAM DESCRIPTION

2.1 GENERAL OVERVIEW

The FRIEDMAN program calculates Friedman's T-statistic for data sets with a maximum of 500 blocks and 30 treatments, allowing for only one observation per treatment/block combination. This is done via the main calling program, MAIN, and five subroutines: INPUT, SORT, XRANK, TSTAT, and OUTPUT. The FRIEDMAN program utilizes an execution time format, providing for a great deal of flexibility in the configuration of the experimental data set. The program is written in FORTRAN IV-H and is currently implemented on the AS/3000 computer in the Earth Observations Data Laboratory, NASA Johnson Space Center.

3. MAIN

The MAIN program is the main routine of FRIEDMAN. This component reads from the terminal the number of card images per block, the number of treatments, and the execution time format. Subsequently, it calls the subroutines that actually calculate the T-statistic.

CALLING PROCEDURE: Not applicable.

INPUT PARAMETERS: NC, NT, FMT

OUTPUT PARAMETERS: M

REFERENCED BY: Not applicable.

SUBPROGRAMS REFERENCED: INPUT, SORT, XRANK, TSTAT, OUTPUT

INPUT/OUTPUT DEVICES: Unit 16 (terminal), Unit 6 (disk, LRECL = 133, BLKSIZE = 133)

LOCAL VARIABLES: The following abbreviations are used in tables throughout this document.

A = alphanumeric
I = integer
R = real

Local variables for the MAIN program are as follows:

Name	Type	Dimension	Description
XINPUT	R	500 × 30	Input data set.
RANK	R	500 × 30	Array of ranked data.
R	R	1 × 30	Sum of ranks for each treatment.

Name	Type	Dimension	Description
DEV	R	1 × 30	Array of squared deviations from the expected value of the treatment sum of ranks.
WORK	R	30	Working storage.
NC	I	1	Number of card images per block.
NT	I	1	Number of treatments.
M	I	1	Number of blocks.
FMT	A	80	Format of XINPUT.

3.1 INPUT

The INPUT subroutine reads the experimental data set from the disk by reading in all treatments for a fixed block. That is, treatments are arranged horizontally and blocks are arranged vertically. (See figure 1 for data configuration.)

CALLING PROCEDURE: CALL INPUT(XINPUT,FMT,M,NT)

INPUT PARAMETERS: FMT, M, NT

OUTPUT PARAMETERS: XINPUT

REFERENCED BY: The INPUT subroutine is referenced by the driver routine MAIN.

SUBPROGRAM REFERENCED: None.

INPUT/OUTPUT DEVICES: Unit 5, the disk containing the experimental data set

LOCAL VARIABLES:

Name	Type	Dimension	Description
XINPUT	R	M × NT	Input data set.
M	I	1	Number of blocks.
NT	I	1	Number of treatments.
FMT	A	80	Format of XINPUT.

$$
\begin{array}{ccccc}
x_{1,1} & x_{1,2} & x_{1,3} & \cdots & x_{1,k_1} \\
x_{1,k_1+1} & \cdots & \cdots & \cdots & x_{1,k2} \\
\vdots & & & & \\
\cdots & \cdots & \cdots & \cdots & x_{1,n} \\
x_{2,1} & x_{2,2} & x_{2,3} & \cdots & x_{2,k1} \\
x_{2,k_1+1} & \cdots & \cdots & \cdots & x_{2,k2} \\
\vdots & & & & \\
\cdots & \cdots & \cdots & \cdots & x_{2,n} \\
\vdots & & & & \\
x_{m,1} & x_{m,2} & x_{m,3} & \cdots & x_{m,k_1} \\
x_{m,k_1+1} & \cdots & \cdots & \cdots & x_{m1,k2} \\
\vdots & & & & \\
\cdots & \cdots & \cdots & \cdots & x_{n,n}
\end{array}
$$

Index I = block; index J = treatment

Figure 1.- Input data configuration.

3.2 SORT

The SORT subroutine sorts the treatment values for each block in ascending order. The algorithm used is a Shell sort (ref. 2).

CALLING PROCEDURE: CALL SORT(XINPUT,INDEX,I,M,NT)

INPUT PARAMETERS: XINPUT, I, M, NT

OUTPUT PARAMETERS: XINPUT, INDEX

REFERENCED BY: The SORT subprogram is referenced by the driver routine MAIN.

SUBPROGRAMS REFERENCED: None.

INPUT/OUTPUT DEVICES: None.

LOCAL VARIABLES:

Name	Type	Dimension	Description
XINPUT	R	M × NT	On input, the raw experimental data; on output, the sorted experimental data.
INDEX	R	NT	Array recording treatment exchanges that occur during the sort.
I	I	1	Number of current block being sorted.
M	I	1	Number of blocks.
NT	I	1	Number of treatments.
NDELTA	I	1	Variable used to partition each block for sorting.
NSWITC	I	1	0 if no switch is made; 1 if a switch is made.
MAX	I	1	NT - NDELTA.
TEMP	R	1	Temporary storage used when interchanging of treatment values occurs.

3.3 XRANK

The XRANK subroutine, using the sorted experimental data, checks for ties and then assigns ranks. Using the array INDEX, XRANK positions treatment I in column I of array RANK.

CALLING PROCEDURE: CALL XRANK(XINPUT,RANK,INDEX,I,M,NT,WORK)

INPUT PARAMETERS: XINPUT, INDEX, I, M, NT

OUTPUT PARAMETERS: RANK

REFERENCED BY::The XRANK subroutine is referenced by the driver routine MAIN.

SUBPROGRAM REFERENCED: None.

INPUT/OUTPUT DEVICES: None.

LOCAL VARIABLES:

Name	Type	Dimension	Description
XINPUT	R	M × NT	Array of sorted experimental data.
RANK	R	M × NT	Array of ranks.
INDEX	R	NT	Array of treatment exchanges that occur during sort.
I	I	1	Block number.
M	I	1	Number of blocks.
NT	I	1	Number of treatments.
WORK	R	NT	Working storage.
NEQUAL	I	1	Number of treatment values making up a tie.
NFIRST	I	1	Column number of first of a sequence of equal values.
LAST	I	1	Column number of the last of a sequence of equal values.

3.4 TSTAT

The TSTAT subroutine calculates Friedman's T-statistic based upon the contents of the array RANK.

CALLING PROCEDURE: CALL TSTAT(RANK,R,DEV,M,NT,T)

INPUT PARAMETERS: RANK, M, NT

OUTPUT PARAMETERS: R, T

REFERENCED BY: The TSTAT subroutine is referenced by the driver routine MAIN.

SUBPROGRAMS REFERENCED: None.

INPUT/OUTPUT DEVICES: None.

LOCAL VARIABLES:

Name	Type	Dimension	Description
RANK	R	M × NT	Array of the ranks of the experimental data.
R	R	1 × NT	Rank totals for treatments.
Z	R	1	Expected value of R(1,J), J = 1, ..., NT.
DEV	R	NT	$[R(1,J) - Z]^2$
S	R	1	$\sum_J [R(1,J) - Z]^2$
T	R	1	Friedman's T-statistic.

3.5 OUTPUT

The OUTPUT subroutine prints an arbitrary M × N array in the following format:

$$
\begin{array}{llll}
x_{1,1} & x_{1,2} & \cdots & x_{1,8} \\
x_{2,1} & x_{2,2} & \cdots & x_{2,8} \\
\vdots & & & \\
x_{m,1} & x_{m,2} & \cdots & x_{m,8} \\
x_{1,9} & x_{1,10} & \cdots & x_{1,16} \\
x_{2,9} & x_{2,10} & \cdots & x_{2,16} \\
\vdots & & & \\
x_{m,9} & x_{m,10} & \cdots & x_{m,16} \\
\vdots & & & \\
x_{1,n-q} & \cdots & x_{1,n} & \\
x_{2,n-q} & \cdots & x_{2,n} & \\
\vdots & & & \\
x_{m,n-q} & \cdots & x_{m,n} &
\end{array}
$$

where q = MOD(NT,8) - 1.

<u>CALLING PROCEDURE</u>: CALL OUTPUT(S,NO,NV)

<u>INPUT PARAMETERS</u>: S, NO, NV

<u>OUTPUT PARAMETERS</u>: None.

<u>REFERENCED BY</u>: The OUTPUT subroutine is referenced by driver routine MAIN.

<u>SUBPROGRAMS REFERENCED</u>: None.

<u>INPUT/OUTPUT DEVICES</u>: UNIT 6, disk with LRECL = 133, BLKSIZE = 133

LOCAL VARIABLES:

Name	Type	Dimension	Description
S	R	NO × NV	Array to be printed.
NO	I	1	Number of rows.
NV	I	1	Number of columns.
H	I	1	NV ÷ 8.
M	I	1	MOD(NV,8).
K	I	1	Lower limit of each print iteration.
L	I	1	Upper limit of each print iteration.

4. EXAMPLE

Assuming the FRIEDMAN program has been compiled, the following EXEC routine may be used to run FRIEDMAN:

```
&CONTROL OFF
FILEDEF 5 DISK &1 &2 &3 (PERM
FILEDEF 6 DISK FRIEDMAN OUTPUT D
     (PERM LRECL 133 BLKSIZE 133
FILEDEF 16 TERM (PERM
LOAD (CLEAR NOMAP
START
SPOOL E NOH
PR FRIEDMAN OUTPUT D (CC
SPOOL E HOLD
&EXIT
```

Suppose the data set to be used is as shown in figure 2. When the above EXEC routine is executed, the following prompt will appear:

ENTER NUMBER OF CARDS PER BLOCK IN I2 FORMAT.

The user will respond with "SPACE 2." The space is necessary because of the I2 format. Upon carriage return, the prompt

ENTER NUMBER OF TREATMENTS IN I2 FORMAT

will appear. The user should respond with "SPACE 7." Upon carriage return, the final prompt will appear:

ENTER INPUT FORMAT.

The user should respond with

(4F3.0,/,3F3.0).

The output will be spooled to the printer no-hold and should appear as shown in figure 3.

14.17.19.14.	Block 1
10.21. 9.	
13.19.15.14.	Block 2
13.13.11.	
21. 9.19.17.	Block 3
9. 7.19.	

Figure 2.- Data set example.

INPUT DATA

	1	2	3	4	5	6	7
1	14.00000	17.00000	19.00000	14.00000	10.00000	21.00000	9.00000
2	13.00000	19.00000	15.00000	14.00000	13.00000	13.00000	11.00000
3	21.00000	9.00000	19.00000	17.00000	9.00000	7.00000	19.00000

a. Input data.

MATRIX OF RANKS

	1	2	3	4	5	6	7
1	3.50000	5.00000	6.00000	3.50000	2.00000	7.00000	1.00000
2	3.00000	7.00000	6.00000	5.00000	3.00000	3.00000	1.00000
3	7.00000	2.50000	5.50000	4.00000	2.50000	1.00000	5.50000

RANK TOTALS

	1	2	3	4	5	6	7
1	13.50000	14.50000	17.50000	12.50000	7.50000	11.00000	7.50000

b. Ranks and rank totals.

FRIEDMAN T

T = 5.750

c. Friedman's T.

Figure 3.- Output example.

5. REFERENCES

1. Conover, W. J.: Practical Nonparametric Statistics. John Wiley & Sons, Inc., New York (New York), 1971.

2. Merchant, Michael J.: Applied FORTRAN Programming With Standard FORTRAN, WATFOR, WATFIV, and Structured WATFIV. Wadsworth Publishing Company, Inc., Belmont, California, 1971, p. 356.

APPENDIX

FRIEDMAN FORTRAN CODE

```
FILE: FRIEDMAN FORTRAN  A   EODL / JOHNSON SPACE CENTER

C FRIEDMAN'S TEST FOR SEVERAL RELATED SAMPLES
C
C
      DIMENSION FMT(80)
      REAL XINPUT(500,30), RANK(500,30), R(1,30),WORK(30)
      REAL DEV(1,30)
      INTEGER INDEX(30)
C
C
C SET UP PROMPTS AT TERMINAL
C
C
      WRITE(16,90)
   90 FORMAT('ENTER NUMBER OF CARDS PER BLOCK IN I2 FORMAT')
      READ(16,95) NC
   95 FORMAT(I2)
      WRITE(16,100)
  100 FORMAT('ENTER NUMBER OF TREATMENTS IN I2 FORMAT')
      READ(16,105) NT
  105 FORMAT(I2)
      WRITE(16,110)
  110 FORMAT('ENTER INPUT FORMAT')
      READ(16,115) FMT
  115 FORMAT(80A1)
C
C
C COUNT NUMBER OF CARD IMAGES & DETERMINE NUMBER OF BLOCKS
C
C
      M=0
  120 READ(5,125,END=5)
  125 FORMAT(1X)
      M=M+1
      GO TO 120
    5 M=M/NC
C
C
C READ DATA
C
C
      CALL INPUT(XINPUT,FMT,M,NT)
      WRITE(6,130)
  130 FORMAT(1H1,' INPUT DATA')
C
C
C WRITE INPUT DATA
C
C
      CALL OUTPUT(XINPUT,M,NT)
C
C
C SORT TREATMENTS & ASSIGN RANKS
C
C
      DO 20 I=1,M
         CALL SORT(XINPUT,INDEX,I,M,NT)
         CALL XRANK(XINPUT,RANK,INDEX,I,M,NT,WORK)
   20 CONTINUE
      WRITE(6,135)
  135 FORMAT(1H1,' MATRIX OF RANKS')
C
C
C WRITE MATRIX OF RANKS
C
C
      CALL OUTPUT(RANK,M,NT)
      WRITE(6,140)
  140 FORMAT(1H0,///,'  RANK TOTALS')
C
C
C CALCULATE FRIEDMAN'S T STATISTIC
C
C
      CALL TSTAT(RANK,R,DEV,M,NT,T)
C
C
C WRITE RANK TOTALS
C
C
```

```
      CALL OUTPUT(R,1,NT)
      WRITE(6,145)
  145 FORMAT(1H1,' FRIEDMAN T')
C
C
C WRITE FRIEDMAN'S T STATISTIC
C
C
      WRITE(6,150) T
  150 FORMAT(1H0,//,'   T = ',F7.3)
      STOP
      END
C
C
C SUBROUTINE INPUT(XINPUT,FMT,M,NT)
C
C
      SUBROUTINE INPUT(XINPUT,FMT,M,NT)
      REAL XINPUT(M,NT),FMT(80)
      REWIND 5
      DO 10 I=1,M
   10   READ(5,FMT)  (XINPUT(I,J),J=1,NT)
      RETURN
      END
C
C
C SUBROUTINE SORT(XINPUT,INDEX,I,M,NT)
C
C
      SUBROUTINE SORT(XINPUT,INDEX,I,M,NT)
      DIMENSION XINPUT(M,NT), INDEX(NT)
C
C
C INITIALIZE INDEX
C
C
      DO 10 J=1,NT
   10   INDEX(J)=J
      NDELTA=NT
C
C
C BEGIN SORT
C
C
   20 NDELTA=NDELTA/2
      IF (NDELTA .LT. 1) GO TO 50
   30 NSWITC=0
      MAX=NT-NDELTA
      DO 40 J=1,MAX
        IF (XINPUT(I,J) .LE. XINPUT(I,J+NDELTA)) GO TO 40
        TEMP=XINPUT(I,J)
        ITEMP=INDEX(J)
        XINPUT(I,J)=XINPUT(I,J+NDELTA)
        INDEX(J)=INDEX(J+NDELTA)
        XINPUT(I,J+NDELTA)=TEMP
        INDEX(J+NDELTA)=ITEMP
        NSWITC=1
   40 CONTINUE
      IF (NSWITC .EQ. 0) GO TO 20
      GO TO 30
   50 RETURN
      END
C
C
C SUBROUTINE XRANK(XINPUT,RANK,INDEX,I,M,NT,WORK)
C
C
      SUBROUTINE XRANK(XINPUT,RANK,INDEX,I,M,NT,WORK)
      REAL XINPUT(M,NT), RANK(M,NT), WORK(NT)
      INTEGER INDEX(NT)
C
C
C ASSIGN INITIAL RANKS & STORE IN ARRAY WORK
C
C
      DO 10 J=1,NT
   10   WORK(J)=FLOAT(J)
      N=NT-1
      NEQUAL=1
```

```
C
C
C CHECK FOR TIES & FIND MEAN RANK WHEN DETECTED
C
C
      DO 20 K=1,N
        IF (XINPUT(I,K) .NE. XINPUT(I,K+1)) GO TO 30
        NEQUAL=NEQUAL+1
        GO TO 20
   30   IF (NEQUAL .GT. 1) GO TO 40
        GO TO 20
   40   NFIRST=K-NEQUAL+1
        INC=NEQUAL-1
        LAST=NFIRST+INC
        SUM=0.
        DO 50 L=NFIRST,LAST
   50     SUM=SUM+WORK(L)
        DO 60 LL=NFIRST,LAST
   60     WORK(LL)=SUM/NEQUAL
        NEQUAL=1
   20 CONTINUE
C
C
C ASSIGN FINAL RANKS TO TREATMENTS
C
C
      DO 70 MM=1,NT
   70   RANK(I,INDEX(MM))=WORK(MM)
      RETURN
      END
C
C
C SUBROUTINE OUTPUT(S,NO,NV)
C
C
      SUBROUTINE OUTPUT(S,NO,NV)
      INTEGER H
      REAL S(NO,NV)
      H=NV/8
      IF (H .EQ. 0) H=1
      M=MOD(NV,8)
      K=1
      DO 10 KK=1,H
        WRITE(6,105)
  105   FORMAT(1X,//)
        L=K+MIN0(NV-1,7)
          WRITE(6,110) (N,N=K,L)
  110     FORMAT(1X,3X,8(8X,I2,5X))
        WRITE(6,120)
  120   FORMAT(1X,/)
        DO 30 I=1,NO
   30     WRITE(6,130) I,(S(I,J),J=K,L)
  130     FORMAT(1H0,I3,8(3X,F12.5))
        K=L+1
   10 CONTINUE
      IF (M .EQ. 0 .OR. NV .LE. 8) GO TO 60
      L=K+M-1
      WRITE(6,140) (I,I=K,L)
  140 FORMAT(1H0,///,4X,8(8X,I2,5X))
      WRITE(6,150)
  150 FORMAT(1X,/)
      DO 50 I=1,NO
   50   WRITE(6,130) I,(S(I,J),J=K,L)
   60 RETURN
      END
C
C
C SUBROUTINE TSTAT(RANK,R,DEV,M,NT,T)
C
C
      SUBROUTINE TSTAT(RANK,R,DEV,M,NT,T)
      REAL RANK(M,NT),R(1,NT)
      REAL DEV(1,NT)
C
C
C FIND RANK TOTALS FOR EACH TREATMENT
C
C
      DO 30 J=1,NT
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
        R(1,J)=0.
        DO 40 I=1,M
   40     R(1,J)=R(1,J)+RANK(I,J)
   30 CONTINUE
C
C
C CALCULATE EXPECTATION OF RANK TOTALS
C
C
      Z=FLOAT(M*(NT+1))/2.
C
C
C CALCULATE SQUARED DEVIATIONS FROM MEAN
C
C
      DO 50 LL=1,NT
   50   DEV(1,LL)=(R(1,LL)-Z)**2
      S=0.
      DO 60 IJ=1,NT
   60   S=S+DEV(1,IJ)
      U=FLOAT(M*NT*(NT+1))
C
C
C CALCULATE T
C
C
      T=(12./U)*S
      RETURN
      END
```


*U.S. GOVERNMENT PRINTING OFFICE: 1982—569-011/546