MET3220C & MET6480
Computational Statistics

Programming – week #3

Dr. Mark Bourassa

TA: David Rock, LOV305A, 644-8577
Office Hours: Tue. 1-2PM & Wed. 2:30-3:30
drock@met.fsu.edu

/usr/lib/sendmail drock@met.fsu.edu < AS3_your_last_name.f90
/usr/lib/sendmail drock@met.fsu.edu < correlate_your_last_name.f90

FORTRAN Tidbit: Procedures AKA Subprograms

- There are two types of procedures: functions and subroutines.
- In general, procedures have zero or more arguments
  - E.g., subprogram1(x1, x2, x3, x4)
  - The variables x1, x2, x3, and x4 are arguments
  - Each procedure must end with the END command
  - Each procedure will cause the program to stop if the program reaches any END command
  - If procedure is not suppose to cause the program to stop, then the program must reach a RETURN command prior to the END.
- Subroutines change the value of one or more of the arguments.
  - Executed with a CALL command. E.g., CALL MEAN(x, ave)
- Functions do not alter any arguments, but return a value.
  - Example: y = mean(x)
- There are several ways to declare procedures.
  - Procedures must be declared in any program that uses them.

FORTRAN90 Example Function

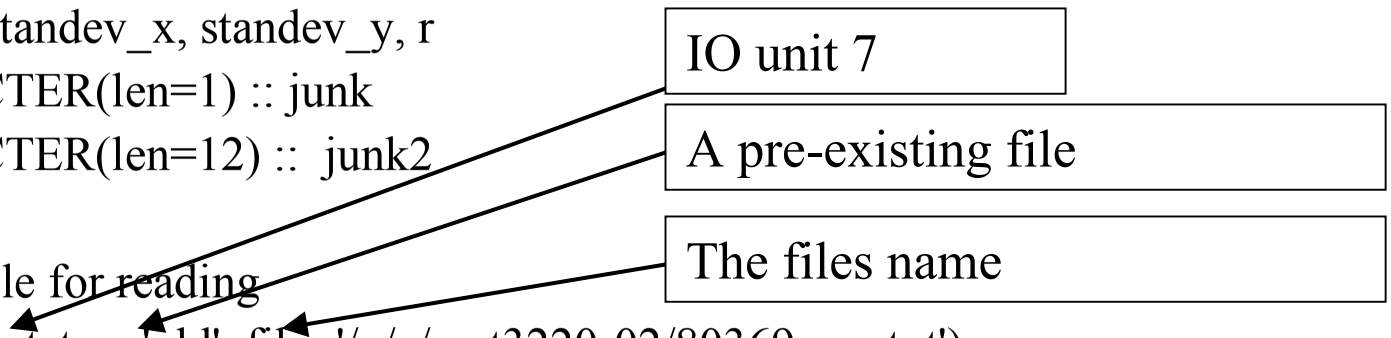
- Consider a subroutine to calculate standard deviation.

```
FUNCTION STANDEV( x, n )
! n   the number of values in array x
! x   array of values for which the standard deviation will be determined
INTEGER :: i_data, n
REAL :: standev, sum_x, sum_x_sqd
REAL, dimension( n ) :: x
sum_x = 0.0
sum_x_sqd = 0.0
DO i_data = 1, n
   sum_x = sum_x + x(i_data)
   sum_x_sqd = sum_x_sqd + x(i_data) ** 2
ENDDO
standev = SQRT( sum_x_sqd – ( sum_x ** 2 ) / n )
RETURN
END FUNCTION STANDEV
```

Review of the Program

```
IMPLICIT NONE
INTEGER :: i_pts, i_skip, n1_pts, n2_pts, n3_pts
REAL, dimension(365) :: tmax1, tmax2, tmax3, tmin1, tmin2, tmin3
REAL :: standev_x, standev_y, r
CHARACTER(len=1) :: junk
CHARACTER(len=12) ::  junk2

! open a file for reading
OPEN( 7, status='old', file='/u/a/met3220-02/80369user.txt')
n1_pts = 366
DO i_skip = 1, 11
  READ(7,*) junk
ENDDO
DO i_pts = 1, n1_pts
  READ(7,'(A12,F4.1,2X,F4.1)') junk2, tmax1(i_pts), tmin1(i_pts)
  print*, tmax1(i_pts), ' ', tmin1(i_pts)
ENDDO
```



Review of the Program


```
CLOSE(7)

!read in data from 2nd city


!read in data from 3rd city

! Correlate minimum and maximum temperature from first city
CALL correlate_your_last_name( tmax1, tmin1, n1_pts, standev_x, standev_y, r )
PRINT*, 'Case 1: ',  standev_x, ' ', standev_y, ' ', r

! After each of the follow correlations, print similar output to above,
! but change the case number.
! Correlate minimum temperature from first city and second city
! Correlate minimum temperature from second city and third city
! Correlate minimum temperature from second city and third city
! Repeat last three steps for daily maximum temperatures for each city

END PROGRAM AS3_your_last_name
```