Chapter 39
Handling Missing Data in Statistics Calculations

Jupiter and beyond the infinite
Stanley Kubrick - 2001: A Space Odyssey

39.1 Introduction

In this chapter we look at a case study of processing the Met Office historic data files and generating statistics accommodating missing data values.

Several steps are involved

- a program to download and save the data files locally
- a sed script to convert the missing values.
- a modified statistics module that will process and report on missing values.
- a module that encapsulates the Met Office station data information.
- a program that actually does the calculations and generates the summary information.
- a site description module that encapsulates the site information.
- a program to generate the site information summary data.

39.2 Example 1: Program to Download and Save the Data Files Locally

This is a C# program. We have programs in Python and Java on our web site that have the same functionality.

Here is the Met Office web address.

```
http://www.metoffice.gov.uk/public/weather/
climate-historic/#?tab=climateHistoric
```

I. Chivers and J. Sleightholme, *Introduction to Programming with Fortran*,
https://doi.org/10.1007/978-3-319-75502-1_39

Here is the program.

```
using System;
using System.Net;
using System.Net.Sockets;
using System.IO;
using System.Text;

class ch3901 {

  static int Main()
  {

    const int n_sites=37;

    string base_address =
    @"http://www.metoffice.gov.uk/pub/"
    +"data/weather/uk/climate/stationdata/";

    string [] station_name =
    {
    "aberporth",        "armagh",
    "ballypatrick",     "bradford",
    "braemar",          "camborne",
    "cambridge",        "cardiff",
    "chivenor",         "cwmystwyth",
    "dunstaffnage",     "durham",
    "eastbourne",       "eskdalemuir",
    "heathrow",         "hurn",
    "lerwick",          "leuchars",
    "lowestoft",        "manston",
    "nairn",            "newtonrigg",
    "oxford",           "paisley",
    "ringway",          "rossonwye",
    "shawbury",         "sheffield",
    "southampton",      "stornoway",
    "suttonbonington", "tiree",
    "valley",           "waddington",
    "whitby",           "wickairport",
    "yeovilton",
    };

    string [] web_address = new string[n_sites];

    string last_part="data.txt";

    string input_string;

    int i;

    // create the web address of each file

    for (i=0;i<n_sites;i++)
    {
      web_address[i]=
      base_address+station_name[i]+last_part;
      System.Console.WriteLine(web_address[i]);
    }

    string[] local_data_file =
    {
      "aberporthdata.txt",        "armaghdata.txt",
      "ballypatrickdata.txt",     "bradforddata.txt",
      "braemardata.txt",          "cambornedata.txt",
      "cambridgedata.txt",        "cardiffdata.txt",
      "chivenordata.txt",         "cwmystwythdata.txt",
      "dunstaffnagedata.txt",     "durhamdata.txt",
      "eastbournedata.txt",       "eskdalemuirdata.txt",
      "heathrowdata.txt",         "hurndata.txt",
      "lerwickdata.txt",          "leucharsdata.txt",
      "lowestoftdata.txt",        "manstondata.txt",
      "nairndata.txt",            "newtonriggdata.txt",
      "oxforddata.txt",           "paisleydata.txt",
      "ringwaydata.txt",          "rossonwyedata.txt",
      "shawburydata.txt",         "sheffielddata.txt",
      "southamptondata.txt",      "stornowaydata.txt",
      "suttonboningtondata.txt",  "tireedata.txt",
      "valleydata.txt",           "waddingtondata.txt",
      "whitbydata.txt",           "wickairportdata.txt",
      "yeoviltondata.txt"
    };

    StreamWriter output_file;

    for (i=0;i<n_sites;i++)
    {
      // create the web addresses

      HttpWebRequest  httpwreq   = (HttpWebRequest)
      WebRequest.Create(web_address[i]);

      // set up connection

      HttpWebResponse httpwresp  = (HttpWebResponse)
      httpwreq.GetResponse();

      // set up input stream

      StreamReader input_stream = new
        StreamReader
        (httpwresp.GetResponseStream(),Encoding.ASCII);

      // read the whole file

      input_string=input_stream.ReadToEnd();

      // create the output file

      output_file =
      File.CreateText("before_"+local_data_file[i]);

      output_file.WriteLine(input_string);

      input_stream.Close();
      output_file.Close();
    }
    return(0);
  }}
```

39.3 Example 2: The Sed Script and Command File That Converts the Missing Values

Here is an extract from one of the Met Office station files.

```
yyyy   mm   tmax    tmin      af    rain     sun
            degC    degC    days      mm   hours
1959    1    4.5    -1.9      20    ---     57.2
1959    2    7.3     0.9      15    ---     87.2
1959    3    8.4     3.1       3    ---     81.6
1959    4   10.8     3.7       1    ---    107.4
1959    5   15.8     5.8       1    ---    213.5
1959    6   16.9     8.2       0    ---    209.4
1959    7   18.5     9.5       0    ---    167.8
1959    8   19.0    10.5       0    ---    164.8
1959    9   18.3     5.9       0    ---    196.5
1959   10   14.8     7.9       1    ---    101.1
1959   11    8.8     3.9       3    ---     38.9
1959   12    7.2     2.5       3    ---     19.2
1960    1    6.3     0.6      15    ---     30.7
1960    2    5.3    -0.3      17    ---     50.2
1960    3    8.2     2.4       4    ---     73.9
1960    4   11.2     2.6       7    ---    146.8
1960    5   15.4     6.5       2    ---    153.9
1960    6   18.5     8.2       0    ---    225.6
1960    7   16.0     9.3       0    ---    111.3
1960    8   16.5     9.4       0    ---    119.2
1960    9   15.0     7.9       0    ---    120.3
1960   10   12.0     5.3       5    ---     ---
1960   11    8.8     2.9       5    ---     37.3
1960   12    5.9     0.4      13    ---     33.9
1961    1    5.4     0.2      11   144.8     31.0
1961    2    8.7     2.9       2   112.5     45.2
1961    3   10.2     2.1      10    77.2    102.6
1961    4   11.9     5.0       1   130.7     83.9
1961    5   ---     ---     ---     66.3    173.7
1961    6   ---      7.4    ---     66.1    190.6
1961    7   16.7     8.2       0   141.1    149.2
1961    8   16.8    10.1       0   149.5    106.6
1961    9   17.4     9.3       0   134.8     79.7
```

The Met Office uses

```
---
```

to indicate a missing value. One way of processing the missing values is to convert the

```
---
```

into a number that cannot occur in the data.

We convert

```
 --- to
-999
```

in this case study. We use the Unix sed command.

sed (stream editor) is a Unix utility that parses and transforms text, using a simple programming language syntax. sed was developed from 1973 to 1974 by Lee E. McMahon of Bell Labs, and is available today for most operating systems. sed was based on the scripting features of the interactive editor ed. ed and vi are sometimes the only editors one has access to on a Unix system. ed is the command driven component of vi. sed was one of the earliest tools to support regular expressions.

The bibliography has some references to sed material.

We can then read the whole file in and adjust the statistics routines to ignore the −999 data values.

Here is the sed command to do the conversion.

```
s/ ---/-999/g
```

Here is an example of the sed command to convert one of the Met Office data files.

```
sed -f convert.sed before_aberporthdata.txt
                        > aberporthdata.txt
```

The -f means read the sed command from a file. sed will read from the file before_aberporthdata.txt and write the converted output to the file aberporthdata.txt

39.4 Example 3: The Program to Do the Statistics Calculations

The complete solution is made up of three source files.

Here is the source code for the statistics module.

```
module statistics_module

  implicit none

contains

  subroutine calculate_month_averages(x, n, &
    n_months, sum_x, average_x, index_by_month, &
    month_names)


    implicit none

    real, dimension (:), intent (in) :: x
    integer, intent (in) :: n
    integer, intent (in) :: n_months

    real, dimension (1:n_months), &
      intent (inout) :: sum_x
    real, dimension (1:n_months), &
      intent (inout) :: average_x

    integer, dimension (1:n), intent (in) :: &
      index_by_month
    character *9, dimension (1:n_months), &
      intent (in) :: month_names

    integer, dimension (1:n_months) :: n_missing
    integer, dimension (1:n_months) :: n_actual

    integer :: m

    sum_x = 0.0
    average_x = 0.0
    n_missing = 0
    n_actual = 0

    do m = 1, n
      if (x(m)>-98.9) then
        sum_x(index_by_month(m)) &
          = sum_x(index_by_month(m)) + x(m)
        n_actual(index_by_month(m)) &
          = n_actual(index_by_month(m)) + 1
      else
        n_missing(index_by_month(m)) &
          = n_missing(index_by_month(m)) + 1
      end if
    end do

    do m = 1, n_months
      average_x(m) = sum_x(m)/(n_actual(m))
    end do

    print *, ' Summary of actual    missing'
    print *, '            values    values'
    do m = 1, n_months
      print 100, month_names(m), n_actual(m), &
        n_missing(m)
100   format (2x, a9, 2x, i6, 2x, i6)
    end do
  end subroutine calculate_month_averages
end module statistics_module
```

The following Fortran segment

```
    do m = 1, n
      if (x(m)>-98.9) then
        sum_x(index_by_month(m)) &
          = sum_x(index_by_month(m)) + x(m)
        n_actual(index_by_month(m)) &
          = n_actual(index_by_month(m)) + 1
      else
        n_missing(index_by_month(m)) &
```

```
            = n_missing(index_by_month(m)) + 1
        end if
      end do
```

is the code to skip processing of the missing data.

Here is the source code for the Met Office station module.

```
module met_office_station_module

  implicit none

  type station_type

    integer :: year
    integer :: month
    real :: tmax
    real :: tmin
    integer :: af_days
    real :: rainfall
    real :: sunshine

  end type station_type

! Number of stations

  integer, parameter :: n_stations = 37

! Number of lines per station, read in later

  integer, dimension (n_stations) :: nl = 0

! Site names

  character *15, dimension (n_stations) :: &
    site_name = (/ 'aberporth      ', &
    'armagh         ', 'ballypatrick   ', &
    'bradford       ', 'braemar        ', &
    'camborne       ', 'cambridge      ', &
    'cardiff        ', 'chivenor       ', &
    'cwmystwyth     ', 'dunstaffnage   ', &
    'durham         ', 'eastbourne     ', &
    'eskdalemuir    ', 'heathrow       ', &
    'hurn           ', 'lerwick        ', &
    'leuchars       ', 'lowestoft      ', &
    'manston        ', 'nairn          ', &
    'newtonrigg     ', 'oxford         ', &
    'paisley        ', 'ringway        ', &
    'rossonwye      ', 'shawbury       ', &
    'sheffield      ', 'southampton    ', &
    'stornoway      ', 'suttonbonington', &
    'tiree          ', 'valley         ', &
    'waddington     ', 'whitby         ', &
    'wickairport    ', 'yeovilton      ' /)


! Station data file names

  character *23, dimension (n_stations) :: &
    station_data_file_name = (/ &
    'aberporthdata.txt      ', &
    'armaghdata.txt         ', &
    'ballypatrickdata.txt   ', &
    'bradforddata.txt       ', &
    'braemardata.txt        ', &
    'cambornedata.txt       ', &
    'cambridgedata.txt      ', &
    'cardiffdata.txt        ', &
    'chivenordata.txt       ', &
    'cwmystwythdata.txt     ', &
    'dunstaffnagedata.txt   ', &
    'durhamdata.txt         ', &
    'eastbournedata.txt     ', &
    'eskdalemuirdata.txt    ', &
    'heathrowdata.txt       ', &
    'hurndata.txt           ', &
    'lerwickdata.txt        ', &
    'leucharsdata.txt       ', &
    'lowestoftdata.txt      ', &
    'manstondata.txt        ', &
    'nairndata.txt          ', &
    'newtonriggdata.txt     ', &
    'oxforddata.txt         ', &
    'paisleydata.txt        ', &
    'ringwaydata.txt        ', &
    'rossonwyedata.txt      ', &
    'shawburydata.txt       ', &
    'sheffielddata.txt      ', &
    'southamptondata.txt    ', &
    'stornowaydata.txt      ', &
    'suttonboningtondata.txt', &
    'tireedata.txt          ', &
    'valleydata.txt         ', &
    'waddingtondata.txt     ', &
    'whitbydata.txt         ', &
    'wickairportdata.txt    ', &
    'yeoviltondata.txt      ' /)

! cwmystwyth   1959 - 2011
! ringway      1946 - 2004
! southampton  1855 - 2000

! default header line count

  integer, dimension (1:n_stations) :: hl = 7

  integer, parameter :: n_months = 12

  character *9, dimension (1:n_months) :: &
    month_names = (/ 'January  ', 'February ', &
    'March    ', 'April    ', 'May      ', &
    'June     ', 'July     ', 'August   ', &
    'September', 'October  ', 'November ', &
    'December ' /)

contains

  subroutine initialise_station_data()
    implicit none
    integer :: i

!   Braemar, Lowestoft, Nairn, Southampton,
!   Whitby
!   have 8 header lines, as the position of
!   the station moved.

    hl(5) = 8
    hl(19) = 8
    hl(21) = 8
    hl(29) = 8
    hl(35) = 8

!   Next read in the current number of
!   lines per station
!   This changes as the data is collected,
!   and when you
!   run the C# program that gets the files.
!
!   I generate this information using wc on the
!   data files.

    open (unit=100, file='line_count.txt',&
              status='old')

    do i = 1, n_stations
      read (100, 100) nl(i)
100   format (i7)
      nl(i) = nl(i) - hl(i)
      print 110, station_data_file_name(i), &
        nl(i)
110   format (' Station ', a30, ' = ', i6, &
        ' records')
    end do

    close (100)

  end subroutine initialise_station_data

  subroutine skip_header_lines(j)

    implicit none
    integer, intent (in) :: j
    integer :: i

!   Skip header lines

    do i = 1, hl(j)
      read (unit=100, fmt='(a)')
    end do

  end subroutine skip_header_lines

end module met_office_station_module
```

Here is the source code for the driving program.

```
include 'ch3903_statistics_module.f90'
include 'ch3903_met_office_station_module.f90'
```

```
program ch3903

  use met_office_station_module
  use statistics_module

  implicit none

! met office data user defined type

  type (station_type), dimension (:), &
    allocatable :: station_data

! Temporary variables used on the read

  integer :: year
  integer :: month
  real :: tmax
  real :: tmin
  integer :: af_days
  real :: rainfall
  real :: sunshine

! Currently we only calculate the
! rainfall sum and averages.

! real, dimension (1:n_months) :: sum_tmax
! real, dimension (1:n_months) :: sum_tmin
! real, dimension (1:n_months) :: sum_af_days
  real, dimension (1:n_months) :: sum_rainfall
! real, dimension (1:n_months) :: sum_sunshine

! real, dimension (1:n_months) :: average_tmax
! real, dimension (1:n_months) :: average_tmin
! real, dimension (1:n_months) ::
! average_af_days
  real, dimension (1:n_months) :: &
    average_rainfall
! real, dimension (1:n_months) ::
! average_sunshine

! Table to hold the monthly rainfall averages
! for all stations.

  real, dimension (1:n_months, 1:n_stations) :: &
    rainfall_table = 0

  integer :: n_years

  integer :: i, j

  call initialise_station_data()


! Process each station

  do j = 1, n_stations

    print *, ' '
    print *, ' Processing ', &
```

```
      station_data_file_name(j)
    print *, ’ ’

    open (unit=100,&
          file=station_data_file_name(j),&
          status=’old’)

!   skip the header lines before starting to
!   read the data

    call skip_header_lines(j)

!   the number of observations at each station
!   is stored in the nl array.

    allocate (station_data(1:nl(j)))

!   Read in the data for each station

    do i = 1, nl(j)
      read (unit=100, fmt=100) year, month, &
        tmax, tmin, af_days, rainfall, sunshine
100   format (3x, i4, 2x, i2, 2x, f5.1, 3x, &
        f5.1, 3x, i5, 2x, f6.1, 1x, f6.1)
      station_data(i) = station_type(year, &
        month, tmax, tmin, af_days, rainfall, &
        sunshine)
    end do

    close (100)

!   Do the monthly average calculations
!   for each station

    call calculate_month_averages(station_data% &
      rainfall, nl(j), n_months, sum_rainfall, &
      average_rainfall, station_data%month, &
      month_names)

    n_years = station_data(nl(j))%year - &
      station_data(1)%year + 1

    print *, ’ ’
    print *, ’ Start date ’, station_data(1)% &
      year, ’ ’, station_data(1)%month
    print *, ’ ’
    print *, ’ Rainfall monthly averages over’
    print 110, n_years
110 format (’  ~ ’, i5, &
      ’ years          mm    ins’)
    do i = 1, n_months
      print 120, month_names(i), &
        average_rainfall(i), (average_rainfall(i &
        )/25.4)
120   format (2x, a9, 8x, f7.2, 2x, f5.2)
    end do
    print 130, sum(average_rainfall), &
      (sum(average_rainfall)/25.4)
130 format (’  Annual rainfall’, /, &
      '  average         ', f8.2, 2x, f5.2)
    print *, ' '
    print *, ' End date  ', station_data(nl(j))% &
      year, ' ', station_data(nl(j))%month

    rainfall_table(1:n_months, j) &
      = average_rainfall

!   Deallocate the arrays

    deallocate (station_data)

!   move on to next station

  end do

  print *, ' '
  print 140, site_name(1:n_stations)
140 format (37(2x,a7))
  print *, ' '

  do i = 1, n_months
    print 150, rainfall_table(i, 1:n_stations)/ &
      25.4
150 format (37(2x,f7.2))
  end do

end program ch3903
```

Here are the required files.

```
line_count.txt
```

Here is some sample output from running the program. It is a subset of the complete output, which can be found on our web site.

```
 Station        aberporthdata.txt       =    906 records
 Station        armaghdata.txt          =   1962 records
 Station        ballypatrickdata.txt    =    660 records
 Station        bradforddata.txt        =   1302 records

... lines deleted

  Processing aberporthdata.txt

  Summary of actual    missing
             values    values
  January        76        0
  February       76        0
  March          76        0
  April          76        0
  May            76        0
  June           76        0
  July           75        0
  August         75        0
  September      75        0
  October        75        0
  November       75        0
```

```
    December       75       0

    Start date  1941   1

    Rainfall monthly averages over
    ~    76 years          mm    ins
    January            90.71   3.57
    February           62.75   2.47
    March              59.25   2.33
    April              53.98   2.13
    May                57.99   2.28
    June               57.23   2.25
    July               61.94   2.44
    August             73.03   2.88
    September          79.43   3.13
    October           104.40   4.11
    November          107.61   4.24
    December          102.01   4.02
    Annual rainfall
    average           910.35  35.84

    End date   2016   6

... lines deleted


    Processing ballypatrickdata.txt

    Summary of actual    missing
               values    values
    January        28      27
    February       28      27
    March          28      27
    April          28      27
    May            28      27
    June           28      27
    July           27      28
    August         27      28
    September      27      28
    October        26      29
    November       28      27
    December       27      28

    Start date  1961   7

    Rainfall monthly averages over
    ~    56 years          mm    ins
    January           133.76   5.27
    February          108.66   4.28
    March              95.51   3.76
    April              87.12   3.43
    May                81.26   3.20
    June               87.10   3.43
    July               90.06   3.55
    August            104.44   4.11
    September          98.58   3.88
    October           148.92   5.86
    November          146.29   5.76
    December          146.13   5.75
    Annual rainfall
```

```
  average          1327.85  52.28

  End date   2016   6

... lines deleted

... following lines truncated
... to fit page

  aberpor  armagh   ballypa  bradfor

     3.57     3.00     5.27     3.60
     2.47     2.25     4.28     2.78
     2.33     2.27     3.76     2.45
     2.13     2.18     3.43     2.28
     2.28     2.35     3.20     2.34
     2.25     2.45     3.43     2.40
     2.44     2.82     3.55     2.61
     2.88     3.23     4.11     3.01
     3.13     2.71     3.88     2.66
     4.11     3.20     5.86     3.16
     4.24     2.90     5.76     3.48
     4.02     3.19     5.75     3.68
```

39.5 Example 4: Met Office Utility Program

The complete solution is made up of two source files.

Here is the source for the site description module.

```
module site_description_module

  type site_description
    character *15 :: site_name = ’ ’
    character *7 :: easting_1 = ’ ’
    character *7 :: northing_1 = ’ ’
    real :: lat_1 = 0.0
    real :: long_1 = 0.0
    integer :: height_1 = 0
    character *7 :: easting_2 = ’ ’
    character *7 :: northing_2 = ’ ’
    real :: lat_2 = 0.0
    real :: long_2 = 0.0
    integer :: height_2 = 0
    integer :: start_date_month_1 = 0
    integer :: start_date_year_1 = 0
    integer :: end_date_month_1 = 0
    integer :: end_date_year_1 = 0
    integer :: start_date_month_2 = 0
    integer :: start_date_year_2 = 0
    integer :: end_date_month_2 = 0
    integer :: end_date_year_2 = 0
  end type site_description

end module site_description_module
```

Here is the source for the driving program.

```
include ’ch3904_site_description_module.f90’

program ch3904

  use site_description_module

  implicit none

  integer, parameter :: n_stations = 37

! site names

  character *15, dimension (n_stations) :: &
    site_name = (/ ’aberporth      ’, &
    ’armagh         ’, ’ballypatrick   ’, &
    ’bradford       ’, ’braemar        ’, &
    ’camborne       ’, ’cambridge      ’, &
    ’cardiff        ’, ’chivenor       ’, &
    ’cwmystwyth     ’, ’dunstaffnage   ’, &
    ’durham         ’, ’eastbourne     ’, &
    ’eskdalemuir    ’, ’heathrow       ’, &
    ’hurn           ’, ’lerwick        ’, &
    ’leuchars       ’, ’lowestoft      ’, &
    ’manston        ’, ’nairn          ’, &
    ’newtonrigg     ’, ’oxford         ’, &
    ’paisley        ’, ’ringway        ’, &
    ’rossonwye      ’, ’shawbury       ’, &
    ’sheffield      ’, ’southampton    ’, &
    ’stornoway      ’, ’suttonbonington’, &
    ’tiree          ’, ’valley         ’, &
    ’waddington     ’, ’whitby         ’, &
    ’wickairport    ’, ’yeovilton      ’ /)

! Braemar, Lowestoft, Nairn,
! Southampton, Whitby
! have 8 header lines, as the position
! of the station moved.

  type (site_description), dimension (1: &
    n_stations) :: site_details

  integer :: i

  open (unit=10, &
        file=’location_line.txt’,&
        status=’old’)

  do i = 1, n_stations

    site_details(i)%site_name = site_name(i)
    read (unit=10, fmt=100) site_details(i) &
      %easting_1, site_details(i)%northing_1, &
      site_details(i)%lat_1, site_details(i)% &
      long_1, site_details(i)%height_1
100 format (10x, a6, 2x, a7, 7x, f6.3, 5x, f6.3, &
      2x, i3)
  end do
```

```
    close (10)

    open (unit=20,&
          file='third_line.txt',&
          status='old')

  ! Update Braemar

  ! print *,' Braemar'

    read (unit=20, fmt=110) site_details(5) &
      %easting_2, site_details(5)%northing_2, &
      site_details(5)%lat_2, site_details(5)% &
      long_2, site_details(5)%height_2

  110 format (2x, a6, 2x, a6, 7x, f6.3, 5x, f6.3, &
      2x, i3)

    site_details(5)%end_date_month_1 = 4
    site_details(5)%end_date_year_1 = 2005
    site_details(5)%start_date_month_2 = 8
    site_details(5)%start_date_year_2 = 2005

  ! Update Lowestoft

  ! print *,' Lowestoft'

    read (unit=20, fmt=110) site_details(19) &
      %easting_2, site_details(19)%northing_2, &
      site_details(19)%lat_2, site_details(19)% &
      long_2, site_details(19)%height_2

    site_details(19)%end_date_month_1 = 8
    site_details(19)%end_date_year_1 = 2007
    site_details(19)%start_date_month_2 = 9
    site_details(19)%start_date_year_2 = 2007

  ! Update Nairn

  ! print *,' Nairn'

    read (unit=20, fmt=110) site_details(21) &
      %easting_2, site_details(21)%northing_2, &
      site_details(21)%lat_2, site_details(21)% &
      long_2, site_details(21)%height_2

    site_details(21)%end_date_month_1 = 12
    site_details(21)%end_date_year_1 = 1997
    site_details(21)%start_date_month_2 = 1
    site_details(21)%start_date_year_2 = 1998

  ! Update Southampton

  ! print *,' Southampton'

    read (unit=20, fmt=110) site_details(29) &
      %easting_2, site_details(29)%northing_2, &
      site_details(29)%lat_2, site_details(29)% &
      long_2, site_details(29)%height_2

    site_details(29)%end_date_month_1 = 12
    site_details(29)%end_date_year_1 = 1969
    site_details(29)%start_date_month_2 = 1
    site_details(29)%start_date_year_2 = 1970



  ! Update Whitby

  ! print *,' Whitby'

    read (unit=20, fmt=110) site_details(35) &
      %easting_2, site_details(35)%northing_2, &
      site_details(35)%lat_2, site_details(35)% &
      long_2, site_details(35)%height_2

    site_details(35)%end_date_month_1 = 12
    site_details(35)%end_date_year_1 = 1999
    site_details(35)%start_date_month_2 = 1
    site_details(35)%start_date_year_2 = 2000

    close (20)

  ! Start dates

    open (unit=30, &
          file='first_data_line.txt',&
          status='old')

    do i = 1, n_stations
      read (30, fmt=120) site_details(i) &
        %start_date_year_1, site_details(i)% &
        start_date_month_1
  120 format (3x, i4, 2x, i2)
    end do

    close (30)

  ! End dates

    open (unit=40, &
          file='end_data_line.txt',&
          status='old')

    do i = 1, n_stations
      select case (i)
      case (5, 19, 21, 29, 35)
        read (40, fmt=130) site_details(i) &
          %end_date_year_2, site_details(i)% &
          end_date_month_2
      case default
        read (40, fmt=130) site_details(i) &
          %end_date_year_1, site_details(i)% &
          end_date_month_1
  130   format (3x, i4, 2x, i2)
      end select
    end do
    close (40)

    do i = 1, n_stations
      print 140, site_details(i)
  140 format (a15, 2x, a7, 2x, a7, 2x, f6.3, 2x, &
        f6.3, 2x, i3, 2x, a7, 2x, a7, f6.3, 2x, &
        f6.3, 2x, i3, 4(2x,i2,2x,i4))
    end do

  end program ch3904
```

Here are the required files.

```
location_line.txt
third_line.txt
first_data_line.txt
end_data_line.txt
```

Here is sample output. It has been reformatted to fit the printed page.

```
aberporth        224100    252100  52.139  -4.570  133
         0.000   0.000    0   1  1941   6  2016   0     0   0     0
armagh           287800    345800  54.352  -6.649   62
         0.000   0.000    0   1  1853   6  2016   0     0   0     0
ballypatrick     317600    438600  55.181  -6.153  156
         0.000   0.000    0   7  1961   6  2016   0     0   0     0
bradford         414900    435200  53.813  -1.772  134
         0.000   0.000    0   1  1908   6  2016   0     0   0     0
braemar          315200    791400   0.000   0.000  339
   315200   791900 57.006  -3.396  327   1
1959   4  2005   8  2005   6  2016
camborne         162700     40700  50.218  -5.327   87
         0.000   0.000    0   9  1978   6  2016   0     0   0     0
cambridge        543500    260600  52.245   0.102   26
         0.000   0.000    0   1  1959   6  2016   0     0   0     0
cardiff          317600    177300  51.488  -3.187    9
         0.000   0.000    0   9  1977   6  2016   0     0   0     0
chivenor         249600    134400  51.089  -4.147    6
         0.000   0.000    0   1  1951   6  2016   0     0   0     0
cwmystwyth       277300    274900  52.358  -3.802  301
         0.000   0.000    0   1  1959   3  2011   0     0   0     0
dunstaffnage     188100    734000  56.451  -5.439    3
         0.000   0.000    0   6  1971   6  2016   0     0   0     0
durham           426700    541500  54.768  -1.585  102
         0.000   0.000    0   1  1880   6  2016   0     0   0     0
eastbourne       561100     98300  50.762   0.285    7
         0.000   0.000    0   1  1959   6  2016   0     0   0     0
eskdalemuir      323400    602600  55.311  -3.206  242
         0.000   0.000    0   1  1911   6  2016   0     0   0     0
heathrow         507800    176700  51.479  -0.449   25
         0.000   0.000    0   1  1948   6  2016   0     0   0     0
hurn             411700     97800  50.779  -1.835   10
         0.000   0.000    0   1  1957   6  2016   0     0   0     0
lerwick          445300   1139700  60.139  -1.183   82
         0.000   0.000    0  12  1930   6  2016   0     0   0     0
leuchars         346800    720900  56.377  -2.861   10
         0.000   0.000    0   1  1957   6  2016   0     0   0     0
lowestoft        654300    294600   0.000   0.000   25
```

```
   653000   293800 52.483   1.727   18   1
1914   8  2007   9  2007   6  2016
manston          632300    166100  51.346   1.337   49
         0.000   0.000    0   7  1934   6  2016   0     0   0     0
nairn            286900    856800   0.000   0.000    8
   291200   857300 57.593  -3.821   23   1
1931  12  1997   1  1998   6  2016
newtonrigg       349300    530800  54.670  -2.786  169
         0.000   0.000    0   1  1959   6  2016   0     0   0     0
oxford           450900    207200  51.761  -1.262   63
         0.000   0.000    0   1  1853   6  2016   0     0   0     0
paisley          247800    664200  55.846  -4.430   32
         0.000   0.000    0   1  1959   6  2016   0     0   0     0
ringway          381400    384400  53.356  -2.279   69
         0.000   0.000    0   1  1946  10  2004   0     0   0     0
rossonwye        359800    223800  51.911  -2.584   67
         0.000   0.000    0  12  1930   6  2016   0     0   0     0
shawbury         355200    322100  52.794  -2.663   72
         0.000   0.000    0   1  1946   6  2016   0     0   0     0
sheffield        433900    387200  53.381  -1.490  131
         0.000   0.000    0   1  1883   6  2016   0     0   0     0
southampton      442000    112500   0.000   0.000   20
   441600   111200 50.898  -1.408    3   1
1855  12  1969   1  1970   3  2000
stornoway        146400    933200  58.214  -6.318   15
         0.000   0.000    0   7  1873   6  2016   0     0   0     0
suttonbonington  450700    325900  52.833  -1.250   48
         0.000   0.000    0   1  1959   6  2016   0     0   0     0
tiree             99800    744800  56.500  -6.880   12
         0.000   0.000    0   1  1928   6  2016   0     0   0     0
valley           230800    375800  53.252  -4.535   10
         0.000   0.000    0  12  1930   6  2016   0     0   0     0
waddington       498800    365300  53.175  -0.522   68
         0.000   0.000    0   1  1947   6  2016   0     0   0     0
whitby           490400    511400   0.000   0.000   60
   489100   510400 54.481  -0.624   41   9
1961  12  1999   1  2000   6  2016
wickairport      336500    952200  58.454  -3.088   36
         0.000   0.000    0   1  1914   6  2016   0     0   0     0
yeovilton        355100    123200  51.006  -2.641   20
         0.000   0.000    0   9  1964   6  2016   0     0   0     0
```

39.6 Bibliography

Dougherty D., Robbins A., sed and awk, O’Reilly

- One of the classic O’Reilly texts.

Unix in a Nutshell, O'Reilly.

- The classic Nutshell text on Unix. Essential reading for Unix and Linux users.

  The wikipedia entry is a good starting place.

  `https://en.wikipedia.org/wiki/Sed`

39.7 Problem

39.1 Compile and run the examples in this chapter.