GUIDE ON USING THE MENUS IN INSTAT AND THEIR EQUIVALENT IN R

African Maths Initiative

BY

Frederic Ntiremganya

October 07, 2015

TABLE OF CONTENTS

TABLE OF FIGURES

Introduction

This guide is a demonstration of how to use a selection of the menus in Instat and and how to their equivalent using R codes through the climate system developed by African Maths Initiative (AMI). In this guide we start with a quick introduction on reading data and the shape of data and then discuss Instat menus with their equivalent in R functions (methods) using tabular, statistics and graphics format.

The demonstration contains examples of menu dialogues and their equivalent command in R. We describe the Instat menus *manage, graphics, statistics and climatic menus with their equivalent functions (methods) in R.*

Importing Data in Instat

This guide provides explanations/description of how to import data into an Instat worksheet both from the Instat library and from external files. The simplest and easiest way of importing a new dataset into Instat is by *File* ➔ *Open from library*. When the dialogue box appears, choose the required dataset in the worksheet then click *OK*. Your dialogue box should look like this as shown in *Figure1*.

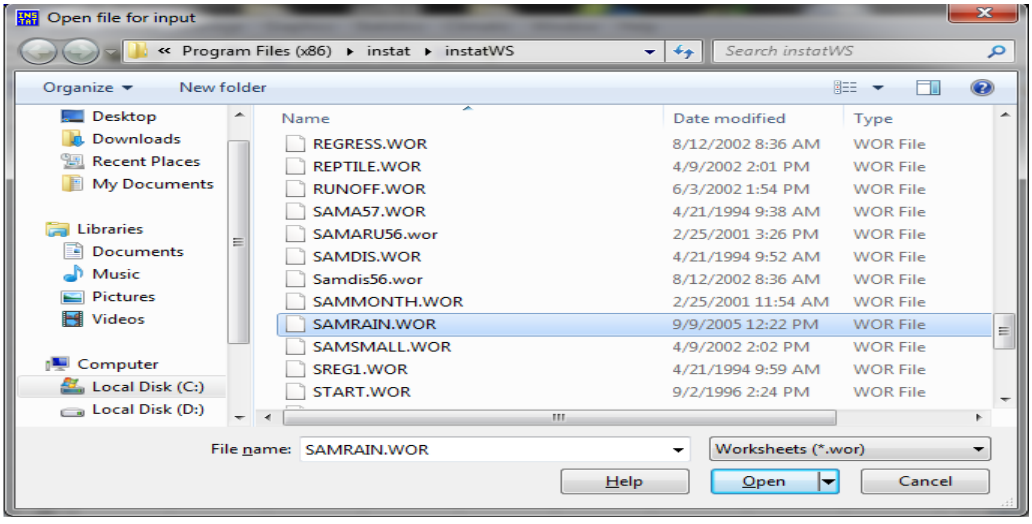

Figure 1 Importing dataset from Instat library

It is possible to import dataset from an external file to Instat. Use *File* ➔ *Open Worksheet (Ctrl + O)*, then locate the file you want to load and click *Open*. The following picture shows how the dialogue looks like in *Figure2*.

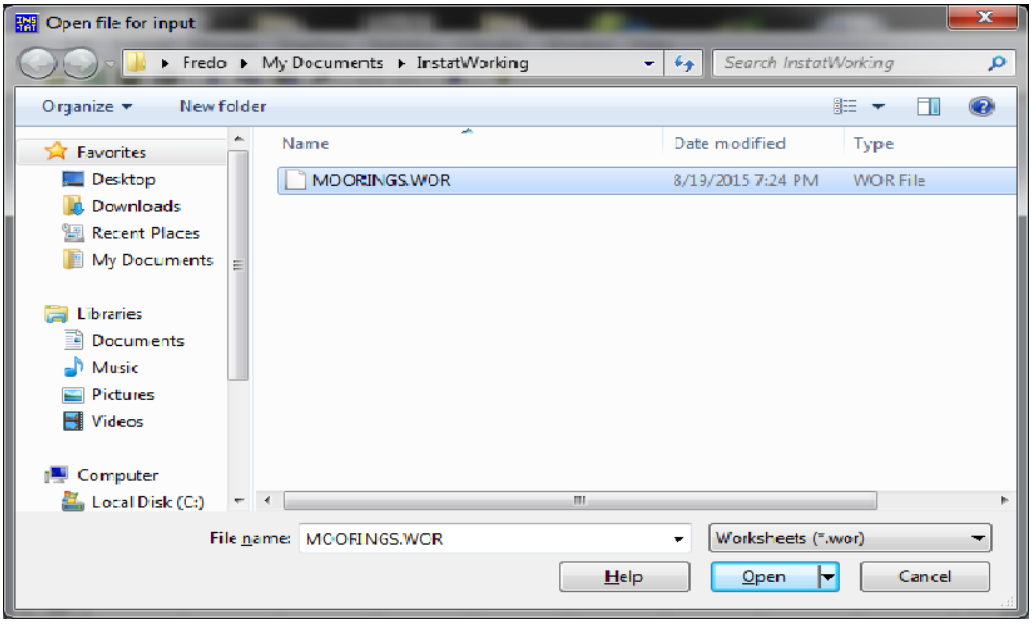

Figure 2 Importing data from external file

Importing Data in R

Importing the data from csv

The first step of the data analysis is to import the data into R. We used different datasets from Statistics in Applied Climatology (SIAC) course to create a single climate object (siac_obj) in order to demonstrate the process. The format of these data sets was csv and we read the datasets into three different data frames as follows:

```
SAMSMALL <- read.csv("C:/Users/Documents/AMI/SIAC/SIAC_docs/R_e_SIAC_docs/Data_sets/SAMSMALL3.csv")

samaru56 <- read.csv("C:/Users/Documents/AMI/SIAC/SIAC_docs/R_e_SIAC_docs/Data_sets/samaru56_2.csv")

samrain <- read.csv("C:/Users/Documents/AMI/SIAC/SIAC_docs/R_e_SIAC_docs/Data_sets/SAMRAIN.csv")
```

SOURCING .R FILES

The code needed to run all the methods for analysis are in .R files. Before we can use them we must source the files using source("..."); if the files are in your working directory you just specify the file name, as below. Otherwise, you must specify the full file path. We source all R files using a sourcing script.

source("SourcingScript.R")

We then created a single climate object of all data frame together called *siac_obj* as follows:

siac_obj <- climate$new(data_tables = list(samaru = samaru56, samsmall = SAMSMALL, samrain = samrain), date_formats = list("%m/%d/%Y","%m/%d/%Y", " "),data_time_periods = list("daily", "daily", "yearly"))

As the above example shows, a climate object can contain multiple data frames, each for a single station. This allows us to carry out analysis on multiple data frames with a single command. It is also possible for the climate object to handle multiple stations given in a single data frame.

Shape of Dataset both in Instat and R

There are two types of seasonal year in the world of farmers leap and non-leap years. To know the leap year, simply check whether a year is divisible by four, and therefore conclude that it is a leap year or not. The 29th February is a single date which occurs every 4 years, which makes that particular year to be divisible by four, hence a leap year. The day of the year in Instat is set to be 366 days by default and assumes a leap year but if the calculation is not for a leap year then the length is decreased to 365.

Since Instat was primarily developed to deal with climate data, part of the design was to only allow it to accept only data that can be possibly recorded in a year. And a year can only have a maximum of 366 days with minimum being 365 days. This is not an issue in R, because R works best with datasets (matrices) and automatically fills in NA for missing values to have an i by j matrix.

Missing Values

Instat allows three missing value codes. They are for missing data (*), for trace precipitation (**) and for non-leap year, February 29^{th,} (***). Trace, which is coded ** is for a very small rainfall amount and Instat treats it as zero for many analyses. R only has a single missing value code, NA (Not Available).

Instat Menus

Manage menu

During data analysis the user might need to add a column in Instat. For instance, they can add a year column by selecting ***Manage ➔ Regular Sequence***. The dialogue will appear and you will need to fill in the first and last year in the dataset and then click ***OK*** as shown in *Figure3.* The year column will be added and you have to rename it manually.

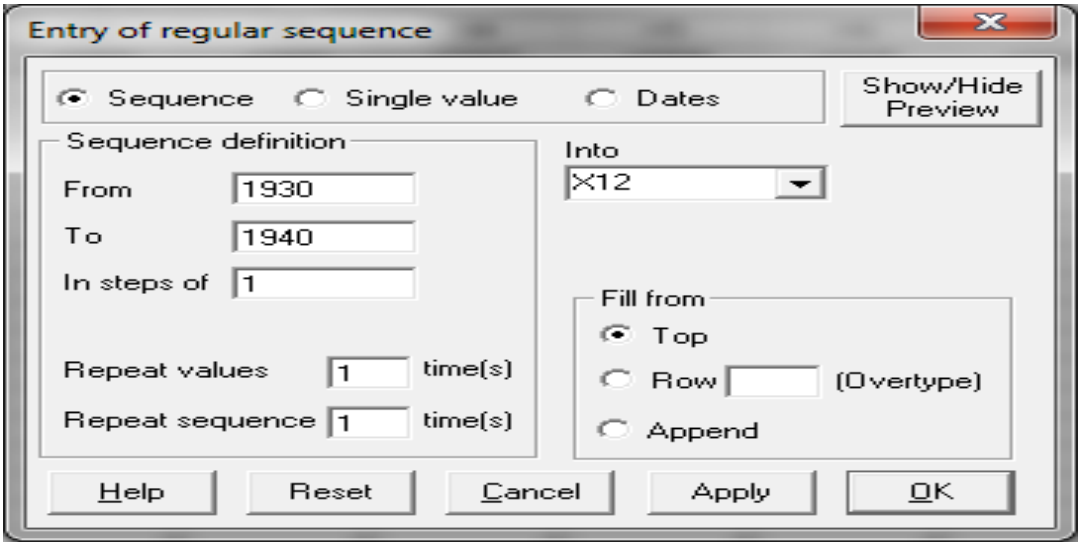

Figure 3 Manage menu showing how to add a column to the data

The same approach is done in R through ***add_year_month_day_cols,*** a climate method developed by AMI. This method can add year, month and day columns if not present in the dataset. It can be used as follows:

siac_obj$add_year_month_day_cols()

We can view the actual data frame as shown below, for one of the original data sets, to see if the year, month and day columns have been appended:

View(head(siac_obj$climate_data_objects[["samaru"]]$data))

	Year	DOY	Rain	Date	Month	Day
1	1928	1	0	1928-01-01	1	1
2	1928	2	0	1928-01-02	1	2

	Year	DOY	Rain	Date	Month	Day
3	1928	3	0	1928-01-03	1	3
4	1928	4	0	1928-01-04	1	4
5	1928	5	0	1928-01-05	1	5
6	1928	6	0	1928-01-06	1	6

Statistics menu

The statistical summaries are given by statistics menu. For instance, you might want descriptive statistics. To do this, choose S***tatistics ➔ Summary ➔ Describe,*** then select the required variable and click ***OK.*** *The de*faults for descriptive statistics are number of observations, minimum, maximum, range, mean and standard deviation. There are more statistics on the same dialogue when clicking on ***additional statistics*** box. The dialogue will appear as shown in *Figure 4* by defaults:

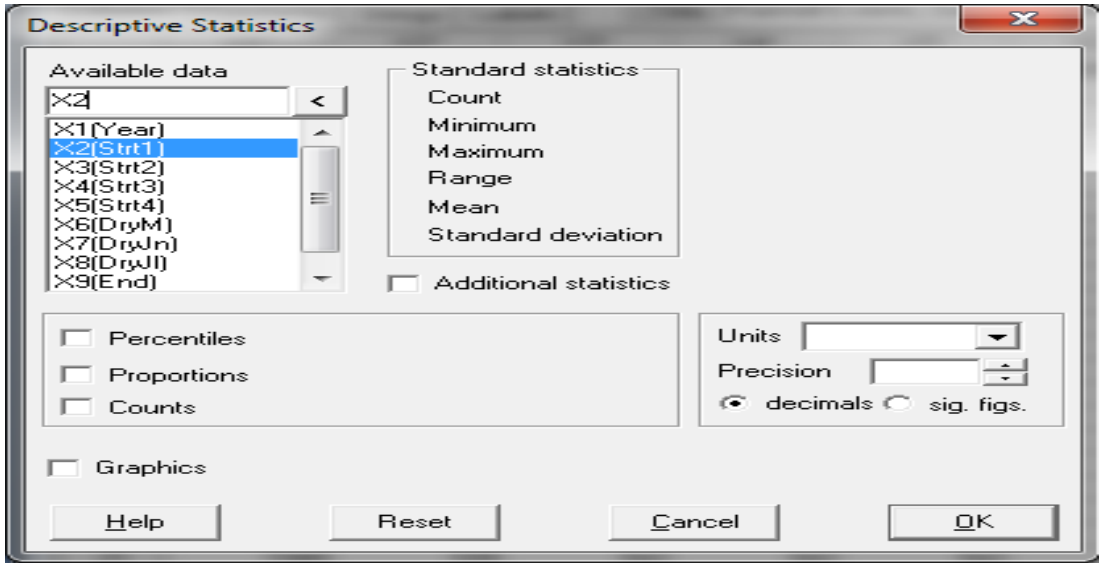

Figure 4 descriptive statistics dialogue for statistical summaries.

Descriptive Statistics Summaries in R

The equivalent of descriptive statistics (mean, minimum, maximum, std.deviation,…) in Instat is done in R. Here the inbuilt function "summary" can be used to compute summaries for a vector.

Summary(samrain$Strt1)

```
> summary(samrain$Strt1)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   94.0   112.5   124.0   126.2   138.5   176.0
```

From the summary command above, we can compute minimum, mean, median, maximum, lower and upper quartile. For other summaries like standard deviation, you can use it's command in R, sd.

Climatic menu

Display of Daily Data

Climatic data is often collected daily. It is important to investigate and summarise raw data. In Instat, the daily data is presented by ***Display Daily*** dialogue. You can display a single year or the whole period of your dataset. To do this, start by opening the data as shown above in ***importing data*** section. The zero rainfall is coded as "--". Choose ***Climatic*** ➔ ***Display Daily*** dialogue, select the year you want to display and then click ***OK.*** This is shown in *Figure 5.*

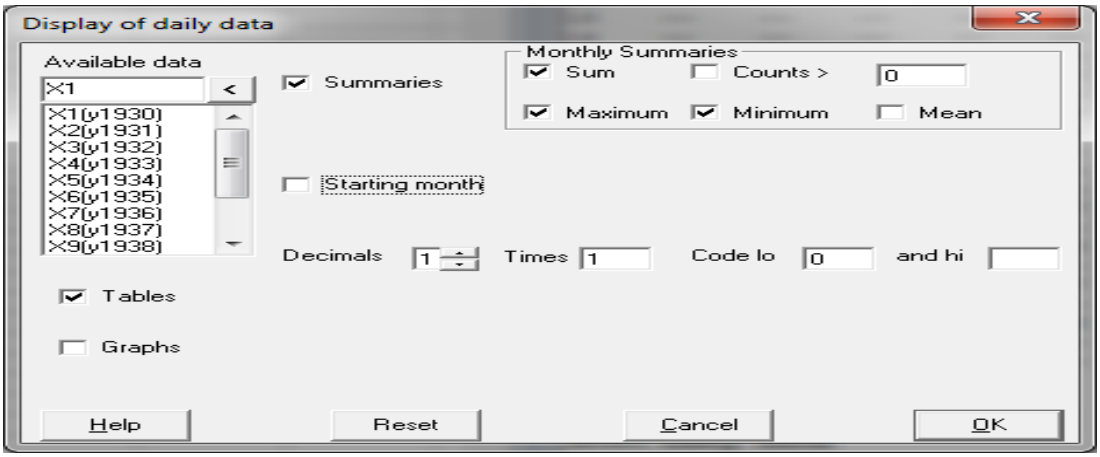

Figure 5 Display daily submenu dialogue for presentation of row data.

Displaying Daily Data in R

The raw climatic daily rainfall data values can be examined in a table. This is performed by display daily menu in Instat and its equivalent is done in R. In Instat, the zero rainfall is coded as -- and missing data value as blank space which is different to R where the missing value data is coded to be NA(Not Available) and the zeros rainfall remain a numeric number, 0.

To view daily data tables in R console use the display_daily method.

Siac_obj$display_daily(summary_names = c("Tot", "Max", "Val>0.85"),decimal_places = 0)

```
$samaru56$`1983`
        Day Jan  Feb Mar  Apr  May  Jun  Jul Aug  Sep Oct  Nov Dec
1         1   0    0   0    0    0    0    0   2    2   0    0   0
2         2   0    0   0    0    0   10   11  35    0   0    0   0
3         3   0    0   0    0    0    4 <NA>  14    0   0    0   0
4         4   0    0   0    0    0    0   10   1    1   0    0   0
5         5   0    0   0    0    0    0    7   1   21   0    0   0
6         6   0    0   0    0    0    0    0   0   11   0    0   0
7         7   0    0   0    0    0    0    1  27    0   0    0   0
8         8   0    0   0    0    0    0    0   0   16   0    0   0
9         9   0    0   0    0 <NA>    0    0   0    0   0    0   0
10       10   0    0   0    0    0 <NA>   18  18    0   0    0   0
11       11   0    0   0    0    4    7    1   0    3   0    0   0
12       12   0    0   0    0   16    0    0   0    0   0    0   0
13       13   0    0   0    0    0    0    0   0    7   0    0   0
14       14   0    0   0    0   17    0    0   0    0   0    0   0
15       15   0    0   0    0    0    7    0   0   12   0    0   0
16       16   0    0   0    0    0    0    6   0   10   0    0   0
17       17   0    0   0    0    0    0    0   0    0   0    0   0
18       18   0    0   0    0    0 <NA>    0  48    5   0    0   0
19       19   0    0   0    0 <NA>    0    0  21    2   0    0   0
20       20   0    0   0    0   17    0    1  15    0   0    0   0
21       21   0    0   0    0    0 <NA>    0   0    0   0    0   0
22       22   0    0   0    0    0    0   18   0    0   0    0   0
23       23   0    0   0    0    2    0    0   6 <NA>   0    0   0
24       24   0    0   0    0    3   24    0   0 <NA>   0    0   0
25       25   0    0   0    0    0    1    0  26    4   0    0   0
26       26   0    0   0    0    1    0 <NA>  25    0   0    0   0
27       27   0    0   0    0    0    0    0   0    0   0    0   0
28       28   0    0   0    0    9   17   27  19    0   0    0   0
29       29   0 <NA>   0    0    0    4    6   0    0   0    0   0
30       30   0 <NA>   0    0    0    0    0   0    0   0    0   0
31       31   0 <NA>   0 <NA>    4 <NA>    0   0 <NA>   0 <NA>   0
32      Tot   0   NA   0   NA   NA   NA   NA 258   NA   0   NA   0
33      Max   0   NA   0   NA   NA   NA   NA  48   NA   0   NA   0
34 Val>0.85   0   NA   0   NA   NA   NA   NA  14   NA   0   NA   0
```

Start of the Rains

The start of the rain is a summary which shows to the farmers when they will start planting their seeds. Consider the start of the rain dialogue for several definitions of the beginning of the season.

Consider the four definitions of start of the rains. For the first define the start to be:

The first day from 1st April with more than 20 mm on a single day, or totaled over 2 consecutive days. This is an example of definition of start of the rains. The user might need to vary any criteria in the definition above and consider an alternative definition, namely: *The first day from 1st May with more than 20 mm on a single day, or totaled*

over 2 consecutive days.

For each definition above, you can generate a new column in your Instat worksheet with each year's start date by choosing ***Climatic*** ➔ ***Events*** ➔ ***Start of the Rains*** dialogue, see the *Figure 6.*

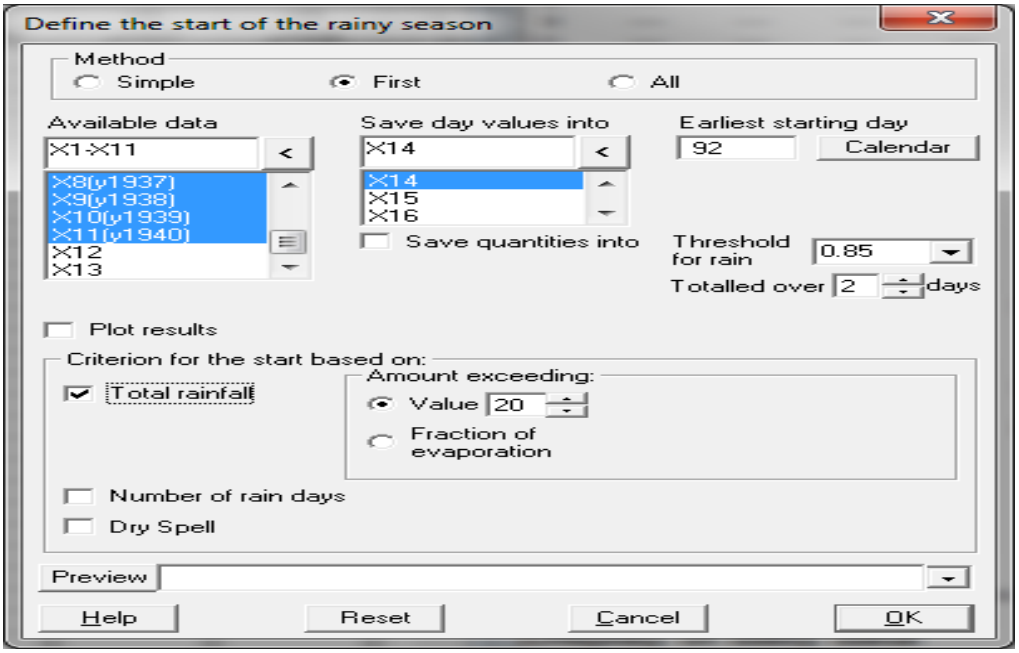

Figure 6 Start of the rains dialogue without conditions

To compute start of the rains for the second definition you only have to change earliest starting day by 1st May (122) rather than 1st April (92).

Consider the 3rd and 4th definitions which involve the dry spell. Let's consider *the definition 1and definition 2 but with the additional condition that there is no 10 day (or longer) dry spell in the next 30 days* respectively.
For conditional definitions (i.e. *with* dry-spell condition), the dialogue box should also includes the dry-spell condition as shown in *Figure 7:*

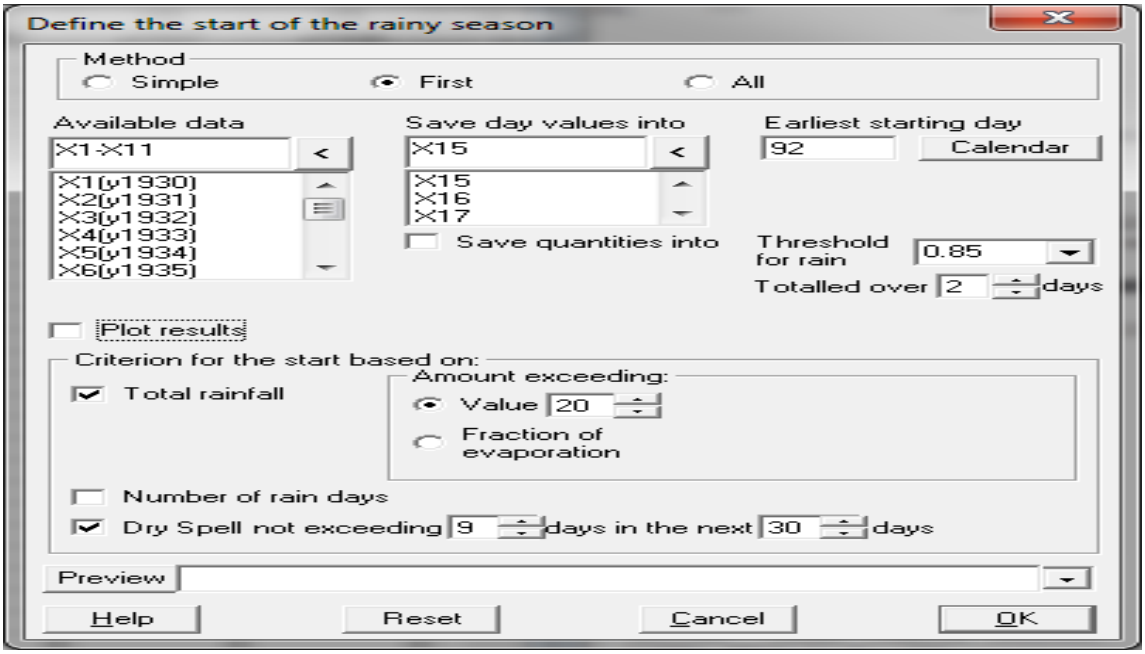

Figure 7 Start of the rains with dry spell condition

How Start of the Rains SubMenu Works in R

With 4 definitions of start of the rains stated above for start of the rain in Instat in mind, we created a start of the rains method that is capable of calculating the start of the rains using any of the definitions.

The command for definition 1 is:

siac_obj$add_start_rain(col_name = "Strt1")

We can view the data using the View command:
View(siac_obj$used_data_objects[["samaru yearly"]]$data)

Or view the first six rows of the data using head function:
View(head(siac_obj$used_data_objects[["samaru yearly"]]$data))

	Year	Date	Total Rain	Number of Rainy Days	Mean Rain per Rainy Day	Strt1
1	1928	1928-01-01	1262.34	85	14.77953	115
2	1929	1929-01-01	1284.18	94	13.62915	126
3	1930	1930-01-01	1044.73	84	12.36774	108
4	1931	1931-01-01	1197.82	76	15.69724	118
5	1932	1932-01-01	1198.16	76	15.72513	115
6	1933	1933-01-01	1311.61	92	14.20707	156

The command for definition 2 is:
siac_obj$add_start_rain (col_name = "Start2 ", earliest_day = 112)

As before, we can view the data using the View command:
View(siac_obj$used_data_objects[["samsmall yearly"]]$data)

Or view the first six rows of the data using head function:
head(siac_obj$used_data_objects[["samsmall yearly"]]$data)

Year	Date	Total Rain	Number of Rainy Days	Mean Rain per Rainy Day	Strt1	Strt2
1930	1930-01-01	1044.73	84	12.36774	108	133
1931	1931-01-01	1197.82	76	15.69724	118	118
1932	1932-01-01	1198.16	76	15.72513	115	115
1933	1933-01-01	1311.61	92	14.20707	156	156
1934	1934-01-01	1076.67	80	13.40775	116	116
1935	1935-01-01	996.43	83	11.92277	134	134

The command for definition 3 is:

We add the start of the rains column to the data with dry spell condition
siac_obj$add_start_rain (dry_spell_condition = TRUE, col_name = "Start3")

As before, we can view the data using the View command:
View(siac_obj$used_data_objects[["samsmall yearly"]]$data)

Or view the first six rows of the data using head function:
head(siac_obj$used_data_objects[["samsmall yearly"]]$data)

The command for definition 4 is:

For definition 4 we specify 112 as the earlist possible day.
siac_obj$add_start_rain (col_name = "Star4)", earliest_day = 112, dry_spell_condition = TRUE)

As the above example, we can view the data using the View command:
View(siac_obj$used_data_objects[["samsmall yearly"]]$data)

Or view the first six rows of the data using head function:
head(siac_obj$used_data_objects[["samsmall yearly"]]$data)

We also produced a method which calculate the end of the rain values to the summary data frame. The following command calculates the end of rain and add the end of the rain values to the summary data frames. It uses the method defaults specified in the report.

siac_obj$add_end_rain()

Use the following command to view a yearly summary:

View(siac_obj$used_data_objects[["samsmall yearly"]]$data)

We can have multiple definitions of end of rain, as we did with Start of rain, by specifying some arguments. For example, the command below will calculate an alternative end of rain.

siac_obj$add_end_rain (earliest_day = 200, evaporation = 6.5, col_name = "End")

Use the following command to view a yearly summary:

View(siac_obj$used_data_objects[["samsmall yearly"]]$data)

	Year	Date	Total Rain	Number of Rainy Days	Mean Rain per Rainy Day	End
1	1930	1930-01-01	1044.73	84	12.36774	210
2	1931	1931-01-01	1197.82	76	15.69724	282
3	1932	1932-01-01	1198.16	76	15.72513	205
4	1933	1933-01-01	1311.61	92	14.20707	290
5	1934	1934-01-01	1076.67	80	13.40775	200
6	1935	1935-01-01	996.43	83	11.92277	220

Extremes

The *extremes submenu* is used to calculate extreme events averaged over a number of days and the day when the event occurred (e.g maximum, minimum or values over a particular threshold). For example, use the ***Climatic ➔ Events ➔ Extremes*** dialogue as shown in *Figure 8*, because we are now interested in when is the maximum daily rainfall during the year, as well as the maximum value itself

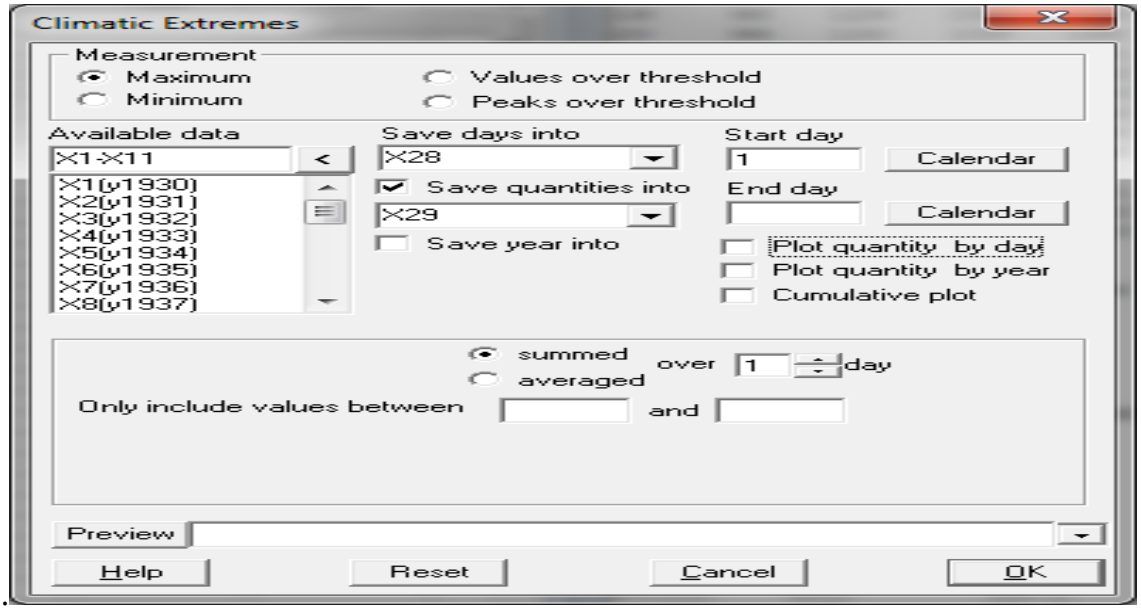

Figure 8 Extremes event dialogue

Extremes menu in R

The extreme dialogue box is from the Climatic menu. Here you can calculate the maximum, minimum or values over a particular threshold. This can be done for a single day. You can also find on which day the event happened. For example, when did the maximum daily rainfall occur each year? This is implemented in R using *extreme_events* method.

The following command calculates the extreme event of interest:
siac_obj$extreme_events(col_name = "extreme")

Use the following command to view the first six rows:

View(head(siac_obj$used_data_objects[[moorings yearly]]$data))

	Year	Date	sum Rain	mean_over_threshold Rain	count_over_threshold Rain	Extreme
1	1928	1928-01-01	1262.34	14.77953	85	77.72
2	1929	1929-01-01	1284.18	13.62915	94	54.10
3	1930	1930-01-01	1044.73	12.36774	84	92.96
4	1931	1931-01-01	1197.82	15.69724	76	64.26
5	1932	1932-01-01	1198.16	15.72513	76	64.52
6	1933	1933-01-01	1311.61	14.20707	92	95.25

Dry Spell in the Rainfall Records

The obvious definition for a dry day is any day with zero rainfall. We first define a dry day and choose to define it as a day with rain less than a certain threshold to avoid any rounding problem which might occur in recording of small amount of rainfall.

Simply choose ***Climatic*** ➔***Events*** ➔ ***Spells*** dialogue in Instat as shown in *Figure 9*. Here the daily records are simply transformed into spell lengths.

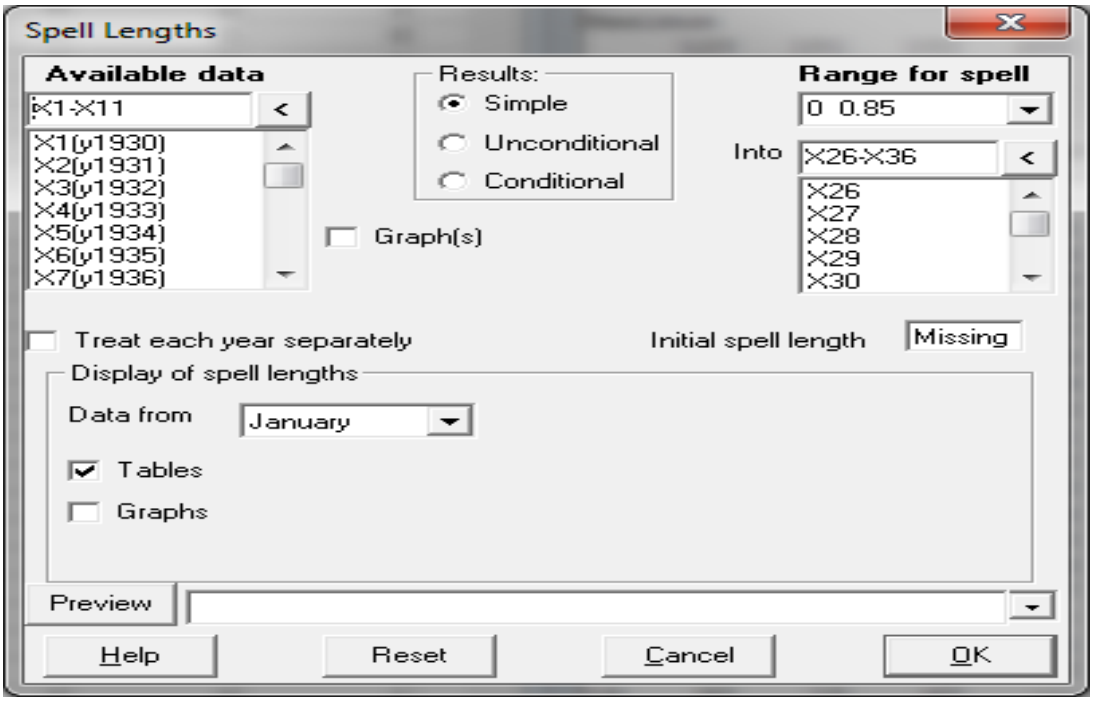

Figure 9 Spells dialogue to transform rainfall data into spell lengths

Having the idea of how spells dialogue works in Instat, we coded *display_spell_length* method in order to return the spell length tables in the R console. This method relies on the spell length values in the daily data. If the daily data does not contain a spell length column, it will call the climate data method *add_spell_length_col*, for each data frame to calculate and add the spell length to the daily data.

The spell length tables can be viewed with the following command:

siac_obj$ display_spell_length ()

The resulting dry spell length table of a single year for one station is as follows:

$`1958`

	Day	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1	1	78	109	137	--	--	2	1	6	2	5	25	55
2	2	79	110	138	1	1	3	--	7	--	--	26	56
3	3	80	111	139	2	--	--	1	8	1	1	27	57
4	4	81	112	140	3	1	1	2	9	2	2	28	58
5	5	82	113	141	4	--	--	3	--	--	3	29	59
6	6	83	114	142	5	1	--	--	1	1	--	30	60
7	7	84	115	143	6	2	1	--	--	--	--	31	61
8	8	85	116	144	7	3	--	--	--	1	1	32	62
9	9	86	117	145	8	4	1	1	--	--	2	33	63
10	10	87	118	146	9	5	--	2	1	--	3	34	64
11	11	88	119	147	10	6	--	3	2	--	4	35	65
12	12	89	120	148	11	7	--	4	--	1	5	36	66
13	13	90	121	149	--	8	--	5	1	--	6	37	67
14	14	91	122	150	1	9	1	6	--	1	7	38	68
15	15	92	123	151	2	10	--	7	1	2	8	39	69
16	16	93	124	152	3	11	1	8	2	3	9	40	70
17	17	94	125	153	4	12	2	--	--	--	10	41	71
18	18	95	126	154	5	13	--	1	1	1	11	42	72
19	19	96	127	155	--	14	1	--	--	2	12	43	73
20	20	97	128	156	--	15	--	1	1	3	13	44	74
21	21	98	129	157	1	--	1	2	--	--	14	45	75
22	22	99	130	158	2	--	2	3	--	--	15	46	76
23	23	100	131	159	3	1	--	--	--	--	16	47	77
24	24	101	132	160	4	2	1	--	--	1	17	48	78
25	25	102	133	161	5	--	2	1	1	--	18	49	79
26	26	103	134	162	6	1	3	--	--	--	19	50	80
27	27	104	135	163	7	--	4	1	--	1	20	51	81
28	28	105	136	164	8	1	--	2	1	2	21	52	82
29	29	106		165	--	2	1	3	2	3	22	53	83
30	30	107		166	1	--	--	4	--	4	23	54	84
31	31	108		167		1		5	1		24		85
32	<NA>											(Overall:	167)
33		108	136	167	11	15	4	8	9	4	24	54	85

Water Balance

The last dialogue in the climatic Events submenu of Instat is for Water Balance. In its simplest form, the ***Climatic ➔ Events ➔ Water*** **balance** dialogue is used similarly to those for ***Spells*** and the ***Start of the rains***, namely as a transformation of the daily data. For example the dialogue, see the *Figure 10* below. Usually we would use a column of data containing the daily evaporation. For simplicity here, assume a single constant value of 5 mm per day. Choose 100mm as the soil capacity.

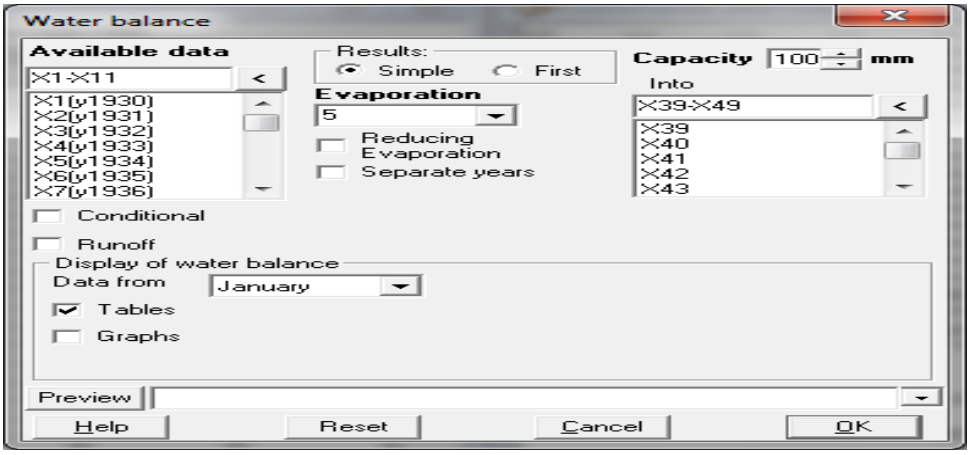

Figure 10 Simple Water Balance dialogue

We replicated the same functionality of *Water Balance menu* from Instat in R using *display_water_balance* method. If the daily data does not contain a water balance column, it will call the climate data method *add_water_balance_col*, for each data frame to calculate and add the water balance to the daily data.

The water balance tables can also be viewed just like we view spell length with the following command:

siac_obj$display_water_balance()

```
$`1982`
   Day Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
1    1   0   0   0   0   0   9   0   0  47   0   0   0
2    2   0   0   0   0   0   2   0   0  46   0   0   0
3    3   0   0   0   0   0   0   0   0  39   0   0   0
4    4   0   0   0   0   0   0   0   0  33   0   0   0
5    5   0   0   0   0   0   0   0   0  56   0   0   0
6    6   0   0   0   0   0   0  18   2  51   0   0   0
7    7   0   0   0   0   0   0  12   0  49   0   0   0
8    8   0   0   0   0   0   0   7   0  43   0   0   0
9    9   0   0   0   0   0   0  22  12  56  34   0   0
10  10   0   0   0   0   4   0  16   6  53  27   0   0
11  11   0   0   0   0   0  10  28   3  55  21   0   0
12  12   0   0   0   0   0   3  22   0  48  14   0   0
13  13   0   0   0   0   0   0  15   0  50   8   0   0
14  14   0   0   0   0   0   0  54   1  44   1   0   0
15  15   0   0   0   0   0   0  47   0  38   0   0   0
16  16   0   0   0   7   0  22  53   0  31   0   0   0
17  17   0   0   0   0   0  15  47  22  25   0   0   0
18  18   0   0   0   0   0   8  40  15  18   0   0   0
19  19   0   0   0   0   0   2  34   8  12   0   0   0
20  20   0   0   0   0  10  21  27   2  21   0   0   0
21  21   0   0   0   0   4  14  21   3  14   0   0   0
22  22   0   0   0   0   8   8  14  48  16   0   0   0
23  23   0   0   0   0   2   2   8  49  10   0   0   0
24  24   0   0   0   0   0   4   1  50   3   0   0   0
25  25   0   0   0   0   0   2   0  47   0   0   0   0
```

26	26	0	0	0	0	0	0	30	42	0	0	0	0
27	27	0	0	0	1	0	0	24	36	0	0	0	0
28	28	0	0	0	0	0	16	18	35	0	0	0	0
29	29	0	NA	0	10	0	9	11	55	0	0	0	0
30	30	0	NA	0	4	0	3	6	48	0	0	0	0
31	31	0	NA	0	NA	16	NA	0	42	NA	0	NA	0

Alternatively, you can view the water balance in the data frame.

View(head(siac_obj$climate_data_objects[["samaru"]]$data))

	Date	spell length	Water Balance	Year	DOY	Rain	Month	Day
1	1928-01-01	NA	0	1928	1	0	1	1
2	1928-01-02	NA	0	1928	2	0	1	2
3	1928-01-03	NA	0	1928	3	0	1	3
4	1928-01-04	NA	0	1928	4	0	1	4
5	1928-01-05	NA	0	1928	5	0	1	5
6	1928-01-06	NA	0	1928	6	0	1	6

Graphics Menu

Plotting lines

The statistics analysis findings can be presented in various ways such as tabular, graphic and written format. Here we describe how to present the findings using ***graphics menu*** in Instat. Use ***Graphics ➔ Plot*** dialogue with y-variable and x-variable, and then click ***OK*** to produce the plot. See the *Figure 11* below.

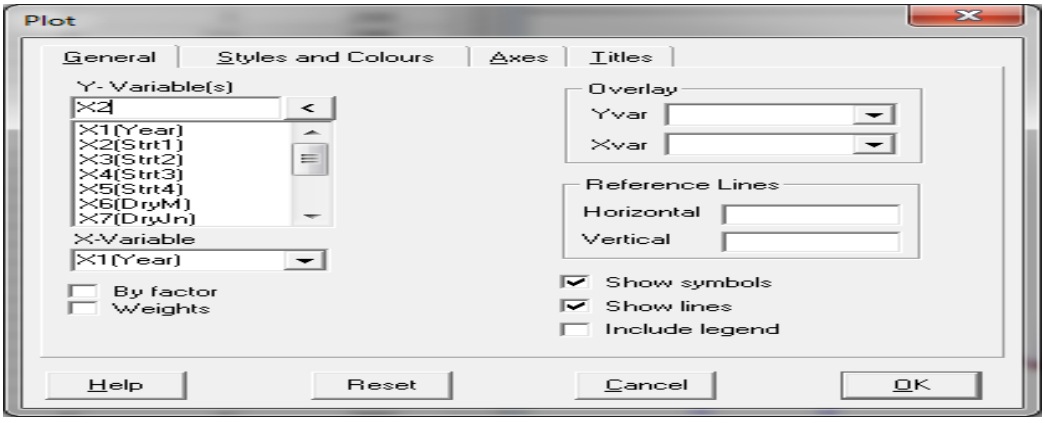

Figure 11 Plot dialogue

To add the title on the graph, just click on the **Titles** tab to add an appropriate title. To get a line plot, tick the **Show lines** checkbox. You might want to specify the type of lines, click on the **Styles and Colours** tab.

We coded the similar function in R with a limitation of plotting yearly variables namely plot_yrar_comparison. The *plot_yearly_comparison* plots a graph of one or more variables on the same plot. Plotting two or more graphs in the same plot gives an idea of comparing the variables when the variables of interest are evaluated at the same x-coordinates. This can be very useful when comparing different variables. For instance, start of the rain for various definitions.

The graph can be viewed with the following command:

ClimObj_obj$plot_yearly_comparison(variables = c("y-variable"))

For instance, run the following command:

siac_obj$plot_yearly_comparison(data_list = list("samrain"),variables = c("Strt1", "Strt2"), type = c("l", "l"),ylabel = "Start of the rain", legend=rep(list(variables), length(variables)),legend.location = rep(list("topleft")), main = "Start of the rain")

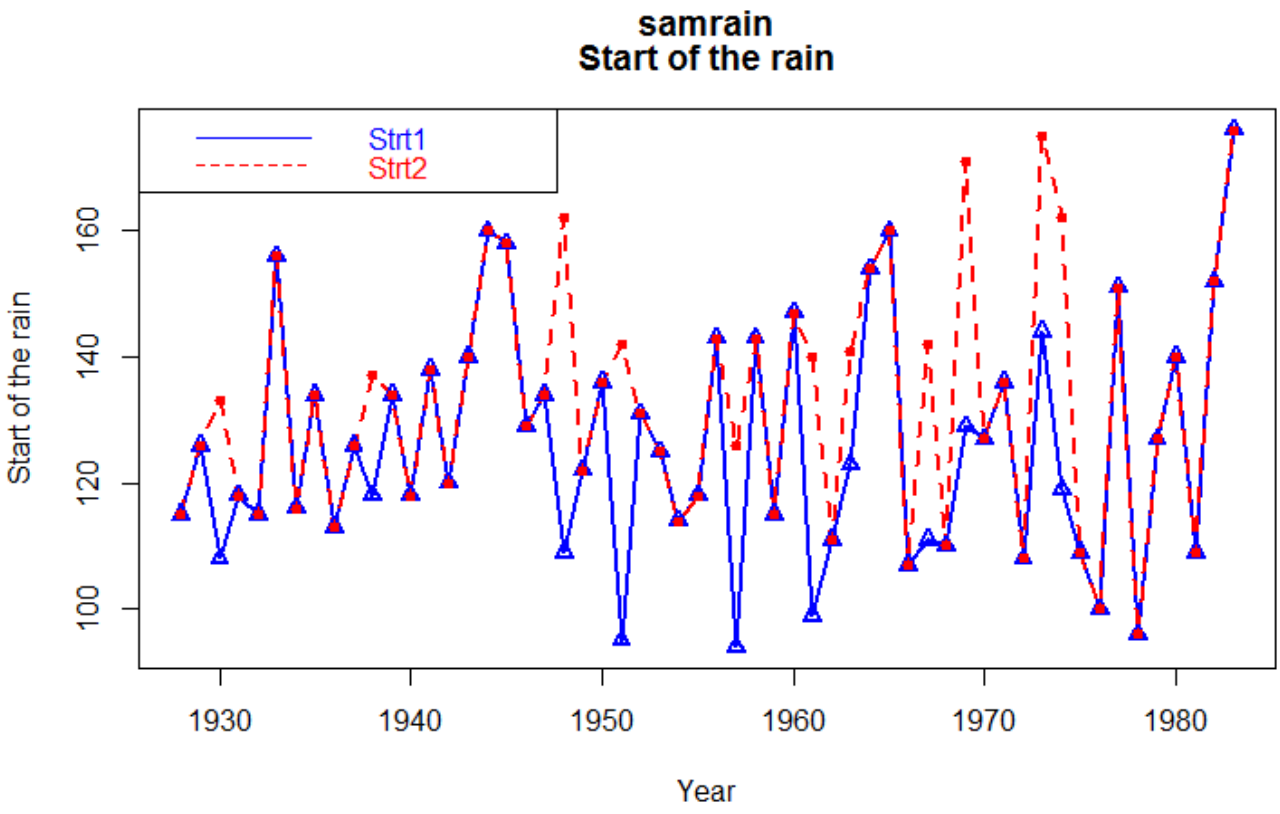

Figure 12 Comparison of two definitions of start of the rains in graph

Box Plot menu

An informative way to illustrate the main aspects of the data is to use a boxplot. Boxplots are used to better understand how values are spaced out in different sets of data. You can find the ***boxplot menu*** as shown in *Figure 13* and complete it.

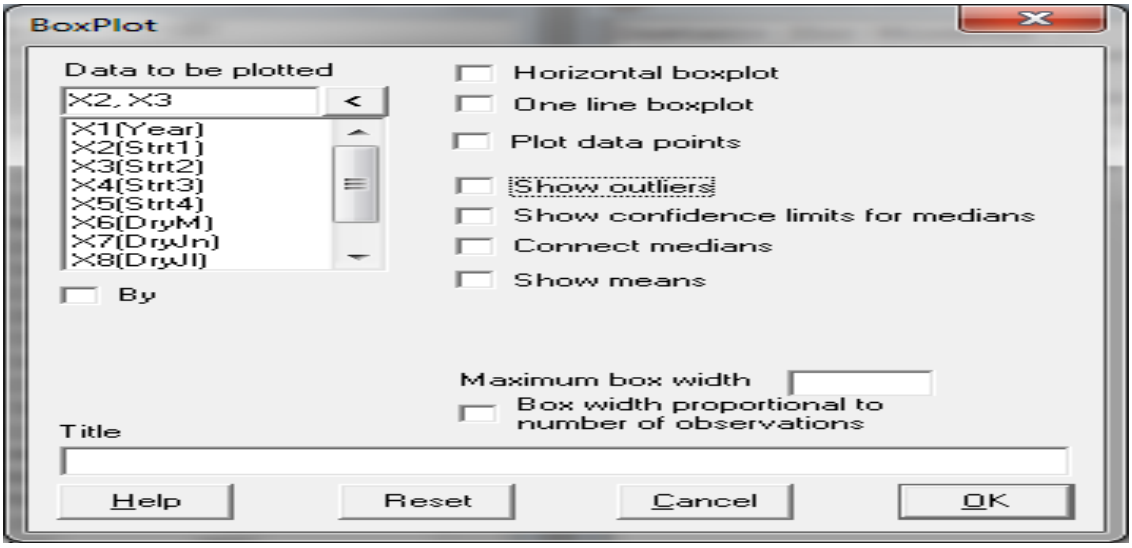

Figure 13 Boxplot menu

The use of *boxplot menu* allows the user to highlight the outliers by just clicking on ***show outlier*** button. It allows also adding some other options like plotting the actual data points, making the box horizontal, etc by clicking on the appropriate button.

In climate object, we have created a method namely *box_jitter* which does the equivalent functionality of *boxplot menu* dialogue from Instat.

The graph can be viewed with the following command:

ClimObj $box_jitter(var = c("variable to plot"), names = c("string names of the plot"))

The following is an example of how **box_jitter** method is used:

siac_obj$box_jitter(data_list = list("samrain"), var = c("Strt1","Strt2","Strt3","Strt4"), names = c("Strt1","Strt2","Strt3","Strt4"))

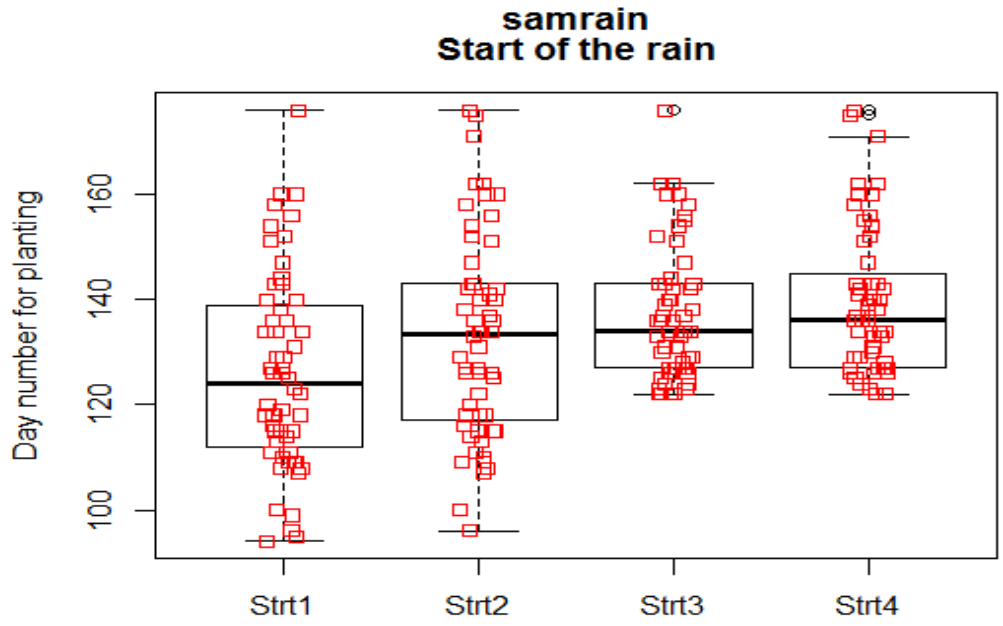

Figure 14 Boxplot with current data points