1 Organizing time series in Matlab structures

A time series is broadly defined as any series of measurements taken at different times. Some basic descriptive categories of time series are 1) long vs short, 2) even time-step vs uneven time-step, 3) discrete vs continuous, 4) periodic vs aperiodic, 5) stationary vs nonstationary, and 6) univariate vs multivariate. These properties as well as the temporal overlap of multiple series, must be considered in selecting a dataset for analysis in this course. You will analyze your own time series in the course. The first steps are to select those series and to store them in structures in a `mat` file. Uniformity in storage at the outset is convenient for this class so that attention can then be focused on understanding time series methods rather debugging computer code to ready the data for analysis. A structure is a Matlab variable similar to a database in that the contents are accessed by textual field designators. A structure can store data of different forms. For example, one field might be a numeric time series matrix, another might be text describing the source of data, etc. In the first assignment you will run a Matlab script that reads your time series and metadata from ascii text files you prepare beforehand and stores the data in Matlab structures in a single `mat` file. In subsequent assignments you will apply time series methods to the data by running Matlab scripts and functions that load the `mat` file and operate on those structures.

How do you put your data put into Matlab structures? A Matlab programmer would do this with a dedicated script that converts data directly from assorted diverse original data files and stores the converted data in fields of structures. Because no programming experience is assumed for the course, you will use a two-step procedure. First is organizing the time series in tab-delimited ascii `txt` files using Microsoft Excel, and the metadata in ascii `txt` files using a text editor. You copy/paste the time series from some source into Excel, and save one version of the Excel file as "text (tab delimited)". You prepare the metadata files with a text editor, such as Notepad++. This lesson describes the preparation of your time series and metadata, and introduces computation and graphics with Matlab. These notes are supplemented by two files: `appendixa.pdf`, which describes Matlab, the toolboxes needed, and how the software might be accessed; and `appendixb.pdf`, which has instructions for formatting and submitting assignments.

1.1 Data types: V1, V2, V3

You must prepare three sets of time series: V1, V2 and V3. Some analyses will treat V1 data as "response" variables and V2 data as "stimulus" variables. In systems terminology, V1 series might be output and V2 series might be input. This arrangement assumes some kind of natural causal relationship. For example, in a dendroclimatology context, precipitation might be considered V2 data and tree-ring indices V1 data because precipitations "affects" tree growth in a causal sense. An important consideration in choosing V2 and V1 time series are this expected causality and that V1 and V2 series must overlap in time sufficiently to examine multivariate relationships. Another important consideration is that the V1 and V2 data should be reasonably stationary in the mean and variance. Obvious trend or strict periodicity (e.g., seasonality) should be removed from the V1 and V2 time series before storing those series.

The V3 data, in contrast, is expected to have some sort of trend that you wish to remove before subsequent analysis. Examples of data choices in dendroclimatology are:

V1 = tree-ring chronologies
V2 = annual precipitation
V3 = measured ring widths for selected cores

Examples of choices in hydrology are:

V1 = gauged total flow for the water year
V2 = water-year total precipitation at stations
V3 = gauged total flow for rivers with anthropogenic influence increasing over time

Examples of choices in climatology are:

V1 = Station Palmer drought severity index
V2 = Station temperature or precipitation
V3 = Global surface temperature

The designation of V2 as "input" and V1 as "output" is nominal. If your own time series do not fall naturally into a physical causal system, you can arbitrarily group one set of time series as "input" series and another group as "output" series. The important thing to keep in mind is that series from V1 will be compared with those in V2 in multivariate analysis (e.g., regression). Choose V1 and V2 so that at least one of the series in V1 is likely to be correlated with at least one of the series in V2.

1.2 Number of series in V1, V2, V3

Include at least 3 series in each of V1, V2 and V2. More than 3 is acceptable, but to avoid clutter in the GUI menus generated by the scripts, include no more than 12 series in any of the structures. I suggest pick a small number of series you really want to understand better rather than a large number of series you might be curious about.

1.3 Length and time-step

The number of observations in a time series is the "length," of the series. Sample length is important for some time series applications. To avoid problems, the time series selected for the course should generally cover at least 30 observations and preferably 50 or more observations. Moreover for bivariate analysis it is necessary that some time series in V1 overlap some series in V2 by at least 30 observations. The series in V3 need not overlap series in V1 and V2. Moreover, it is not necessary that the individual series in V3 overlap in time with one another.

The interval between observations is the "time-step." Time series selected for the course should have a constant time step with an interval of one. Such series are also called "evenly spaced." For example, annual precipitation measured for years 1920, 1921, 1922, 1923 and 1924 has a length of 5 with a constant time step of one year. Tree-ring width chronologies have a natural constant time step of one year. For some time series the time step might be constant but not equal to one year. It is acceptable to have data at time steps other than one year (e.g., second, day, week, quarter), as long as all of your time series (V1, V2, V3) have the same time step and that time-step is constant.

Some types of data can be aggregated in time into constant time steps equal to multiple years. For example, paleoclimatic measurements from ocean cores might represent successive 1000-year blocks of time. In that case, "times" 1, 2, 3, etc., increment by one, but each time is understood to correspond to a block of 1000 years: times 1, 2 and 3 would be the first, second and third 1000 year block of time.

Unevenly spaced time series are not directly amenable to analysis with the methods of this course, and should be avoided. An example of an unevenly spaced time series is the weight of a person measured each time he visits the doctor. Typically these visits are not at regular intervals, despite the recommendations of the doctor.

1.4 Discrete vs continuous

The terms “discrete” and “continuous” as used here refer to the values of the time series, rather than to the sampling frequency. The sense used is therefore “discrete-valued” vs “continuous-valued”. A continuous time series can assume any real number, or any number along a number line or axis. Real numbers can be positive, negative, large or small. They can be whole numbers or fractions. The range may of course be limited depending on the actual data. For example measured annual precipitation cannot be negative, and will fall within some reasonable upper limit imposed by climate. In contrast a discrete time series can assume only some specified subset of numbers. The most extreme case of a discrete time series is a binary series, which must be either 0 or 1. An example of a binary series is whether a tree ring series for a particular tree records a fire or not in each years. Categorical time series include binary series, but can assume some larger number of values, which may be integers coding some trait (e.g., 1=short 2=medium, 3=tall)., which includes all rational and irrational numbers.

Discrete time series also include “counts,” such as the number of events each year. A discrete series can therefore take on few or many different values depending on the type of data. A special category of counting time series is the “point process.” Point processes are times of events, such as earthquakes, lightning strikes, or fires. The events occur as a function of time, and may be rare (few “times” record an event). Some interesting properties of point processes are the intervals, or gaps of time, between events, and the number of events in specified intervals of time. , and. An example is the number of lightning strikes per second over a one-hour period. Point processes can be studied with a particular suite of analytical methods (Brillinger 1994), but are not well suited for the methods used in this course (i.e., avoid them).

For this course it is best to choose continuous time series or discrete time series that can assume a wide range of values. In practice, limitations of measurement systems can blur the distinction between continuous and discrete. For example, air temperature is a continuous time series but in practice can only assume a specified limited number of possible values because because a) the temperature as measured by a common indoor or outdoor thermometer can be read to only about a tenth if a degree, and b) physical processes result in air temperature not falling outside some reasonable observed range. The main points are is to avoid for this course binary series, point processes, or discrete series that can assume only a few possible values.

1.5 Periodic vs aperiodic

A purely periodic time series has statistical properties conditional on the time position of the observation relative to the starting observation time of some physically driven cycle. Monthly climatic series of air temperature observed over multiple years, for example, will be higher in winter than in summer, and will exhibit a well-defined 12-month cycle. Insolation changes associated with the earth’s orbit around the sun drive the cycle. Likewise, hourly air temperature measured over a many days will typically have highs in the afternoon, lows in the early morning, and a 24-hour cycle. The rotation of the earth and the resulting changes in exposure to solar radiation drive the cycle. Such time series can be called “periodic,” or “seasonal,”, in that a significant part of their variation is driven by a recognized cyclical physical processs.

Except for demonstrating periodicity, you should avoid periodic time series in your selection of data for the course. Such data is fine for the V3 dataset, but not for V1 and V2. If you want to use a periodic time series in the V1 or V2 data, you should first adjust the series to remove the periodic component. Adjustment can be by sampling, aggregation, or explicit removal of the seasonal component. The sampling approach is to use samples of the time series at some set phase, or position, of the cycle. For example, instead of using a monthly time series of 1200 values of monthly temperature extending over a 100 year period, you can use only the 100 values

corresponding to the month of January. The sacrifice is that the length of the series to be analyzed is 1/12 the length of the original series

The aggregation approach is to sum or average the seasonal series to remove the seasonal component. For example, instead of monthly air temperature, you could use the air temperature averaged over the 12 months of the year. Or, a hybrid of sampling and aggregation is to use the time series of monthly precipitation averaged over some group of months (e.g., summer mean temperature). The aggregation approach also necessarily results in a reduction of series lengths.

Finally, the seasonal dependence can be statistically removed from the seasonal time series. For example, a time series of monthly air temperature could be expressed as departures from the individual monthly long-term means. This approach has the advantage over sampling and aggregation of retaining the original sample length.

Regardless of how you might prepare remove seasonality, the resulting time series should have a unit time step (time step of one) for use in the course. For the examples of monthly temperature series reduced by sampling or aggregation, the time step would be 1 years, and you could use the calendar year as the time variable (e.g., 1900, 1901, 1902, …). For the example of monthly temperature adjusted by subtracting the long-term mean, you would use a sequential monthly time variable. For example, the times 1-12 would correspond to months Jan-Dec of year 1, times 13-24 to months Jan-Dec of year 2, etc.

1.6 Stationary vs nonstationary

While stationarity has a specific statistical definition that will be discussed later, it is sufficient for practical purposes at this data-selection step to consider as "stationary" any time series that has no obvious temporally-changing mean level. Stationary time series are preferred for V1 and V2 datasets in this course because some the methods that will be applied to the V1 and V2 data assume stationarity. On the other hand, feel free to include series with obvious trend in dataset V3.

1.7 Univariate vs multivariate

Univariate refers to a single time series and multivariate to several time series. This is an introductory course, and as such focuses primarily on univariate methods, with a particular time series method applied to one time series at a time. However, correlation analysis moves to the category of "bivariate" time series (analysis treats two series), and the regression modeling is multivariate in the sense the predictor data for regression is a set of several time series.

1.8 Input format of time series

Organize and save the time series for V1, V2, and V3 in three tab-delimited ascii `txt` files. The easiest way to do this is with Excel, which allows you to "save as" text (tab delimited). Matlab function `geosa1` will subsequently be used to read the tab-delimited data and store it in Matlab structures. Say you are entering or copy/pasting the time series into an Excel spreadsheet. Column 1 should hold the time variable (e.g., the year), and the remaining columns should hold the time series. The time variable in column 1 should increase down the column – in other words, the earliest times are towards the top rows, and the most recent times towards the bottom rows.

Row 1 should contain short (15 or fewer characters) headers, or variable labels. These labels should not have any internal spaces, and should be identical to and in the same order as the labels in the metadata file (see below). The label for the time-variable should also exactly match the time label in row 3 of the metadata file. To avoid complications, include only letters and numbers in these labels. Avoid other characters (e.g., the "underscore"). Once all data have been entered in the `xls` file, save it as a text (tab delimited) `txt` that can be read by `geosa1`.

1.9 Missing data

The data selected for V1, V2, and V3 should be continuous in the sense that there are no internal missing values. If you want to use time series that have missing values, fill in those missing values by some appropriate method before storing the data in V1, V2 or V3.

The block of cells with time series in the `xls` file and tab-delimited `txt` file is a time series matrix. This matrix can have missing data (blanks, or some specified numeric code), but the missing data must be at the ends of the series, not internal. For example, if the block of data covers the years 1801-2006, one series might have valid data just for the portion 1830-1999, another for 1815-2001, etc. The cells without valid data must be filled with a missing-value code, which can be "NaN" (Matlab's missing number indicator), or any numeric value that will not be confused with actual data (e.g., -999.0). `Geosa1` will convert the tab-delimited data to a standard form for use in exercises in the course, and will replace any missing-value code with "NaN".

1.10 File naming

A separate tab-delimited `txt` data file must be prepared for each of the input data sets V1, V2, and V3 described in 1.1. Save those files with names `V1Data.txt`, `V2Data.txt` and `V3Data.txt`.

`Geosa1` will read the tab-delimited `txt` files `V1Data.txt`, `V2Data.txt` and `V3Data.txt`, extract the time series from those files, extract the metadata from the paired metadata files (see below), and store the combined information in a Matlab storage (`mat`) file (.mat file). `Geosa1` will prompt you for your last name, and use your name in naming the output `mat` file. For example, if, when prompted by geosa1 for your last name, you enter "Lincoln", the output file generated and saved by `geosa1` will be "`Lincoln.mat`".

1.11 Supporting information for time series—the metadata

Each of the three tab-delimited time-series files (`V1Data.txt`, `V2Data.txt`, `V3Data.txt`) files) should be accompanied by a `txt` metadata file. The metadata file has additional information on the time series in the data files. We require minimal metadata in this course, and the information is used only in labeling the graphics and data series in menus.

`Geosa1` requires these `txt` metadata files, so you must build them beforehand. Any text editor that produces ASCII output (e.g., notepad++) will do. The metadata `txt` files have a very simple structure, which you must follow exactly or else `geosa1` will fail. Each `txt` file consists of three initial lines followed by one line of information for each time series. The three initial lines apply to all series in the paired ascii-delimited data file, and are:

1 filename of the tab-delimited `txt` file with the time series
2 time step for the data
3 missing value code used in the tab-delimited `txt` file

Subsequent lines, one per time series, give the following information, with elements on a line separated by a dollar sign, "$":

1 short name, for labeling series, 15 or fewer characters
2 long name of series, 40 or fewer chars
3 y-axis label for variable (type of data), 20 fewer characters
4 units of variable, 13 or fewer characters

The short name should be sufficient for you to identify series in labeled plots, and must be exactly the same as the variable labels in the first row of the tab-delimited fie with the corresponding time series (see above). The long name can be more descriptive and can contain internal spaces, and special character, but avoid the underscore ("_") which Matlab uses to mark subscripted test. The y-axis labels and the "units" descriptor should not contain internal spaces or underscores.

1.12 Some examples of time series

The most important summary plot in time series analysis is a simple line plot of the variable of interest as a function of time. This plot is called a "time series plot" or a "time plot." Examples of such plot for a variety of types of time series are shown in Figures 1-11. The plots serve to illustrate some of the points raised in the preceding sections (see captions of figures).

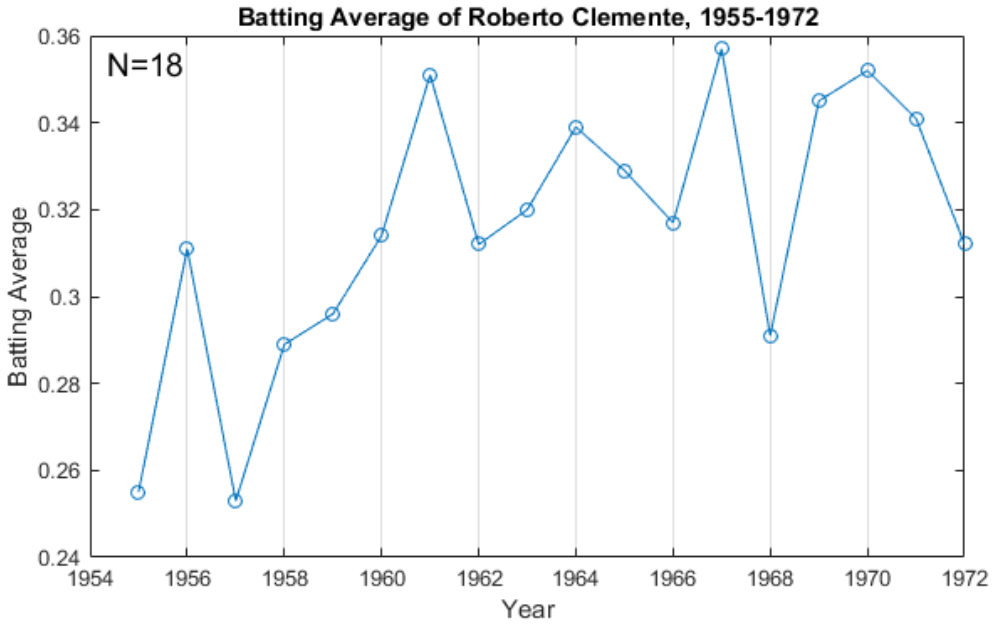

Figure 1. Batting average of major league Hall of Fame outfielder Roberto Clemente. Series is very short (N<30 observations), continuous-valued, and aperiodic, with a constant time step of 1 year and apparent stationarity of mean. Continuous valued series can assume any real number. Batting average is computed as number of hits divided by number of attempts, or at-bats, such that the average cannot fall outside the range 0-1.0. Within that restricted range, the batting average can take on any value (possibilities are actually limited by the number of at-bats). Judgements about stationarity are necessarily subjective, and more so the shorter the time series. Source of data: Baseball-reference.com

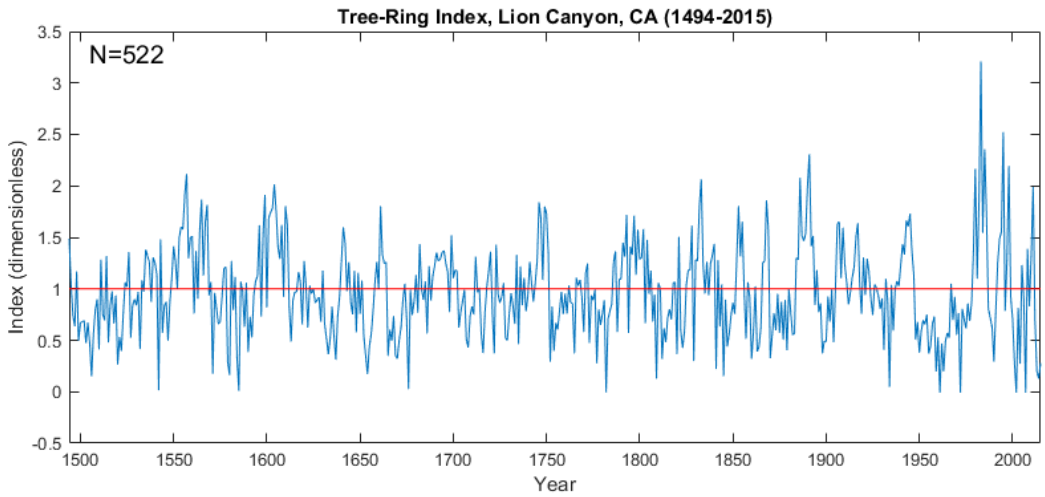

Figure 2. Tree-ring index site chronology for Lion Canyon, California. Index is interpreted as decimal proportion of normal growth, such that an index of 1.0 is "normal." Series is long (N>>50 observations), continuous-valued, and aperiodic, with a constant time step of 1 year and apparent stationarity of mean. "Long" and "short" are subjective judgements, but for purposes of illustrating time series methods in the course, series longer than 50 years are ideal, and longer than 30 years are preferred. Source of data: dendrohydrology project for California Dept. of Water Resources, D. Meko and C. Woodhouse, PIs.

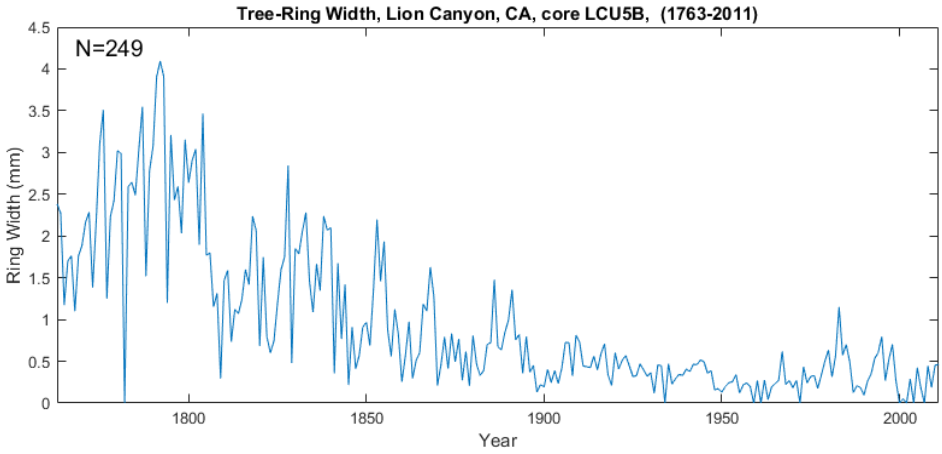

Figure 3. Tree-ring width measurements from a single tree at Lion Canyon, California. Series is long (N>>50 observations), continuous-valued, and aperiodic, and nonstationary, with a constant time step of 1 year. The steady decline in level suggests nonstationarity, or systematic change over time, of mean. The decrease in spread suggests nonstationarity of variance as well. Source of data: dendrohydrology project for California Dept. of Water Resources, D. Meko and C. Woodhouse, PIs.

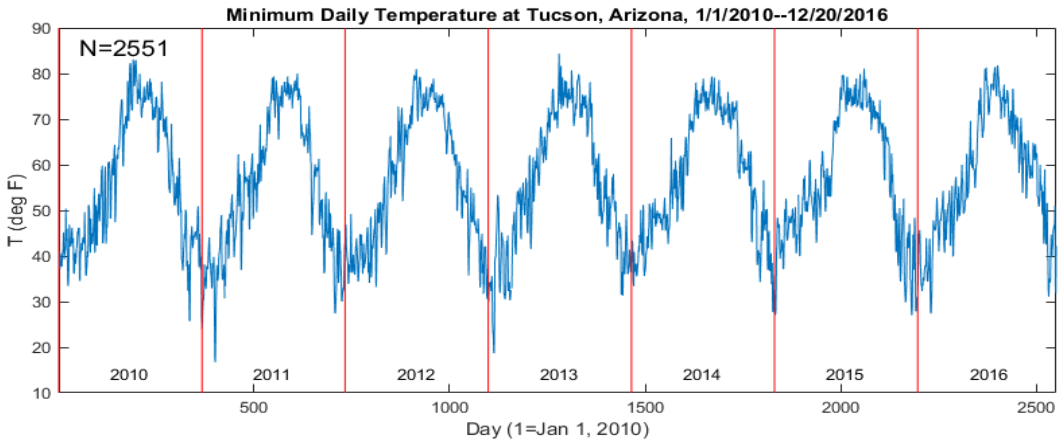

Figure 4. Minimum daily temperatures at Tucson, Arizona, over a period of 7 consecutive years. Series is long (N>>50 observations), continuous-valued, and periodic, with a constant time step of 1 day. Note that days are numbered consecutively from 1, such that the last values is day 2551. The annual cycle, or "period," is obvious: lowest daily minimums occur in mid-winter (red line is Jan 1) and highest daily minimums in mid-summer. The seasonal cycle also makes the series strictly nonstationarity of mean, as the expected mean of the series is conditional on day of year. However, aside from the cycle, no gradual change in mean level is evident. Source of data: PRISM Data Explorer (http://www.prism.oregonstate.edu/).

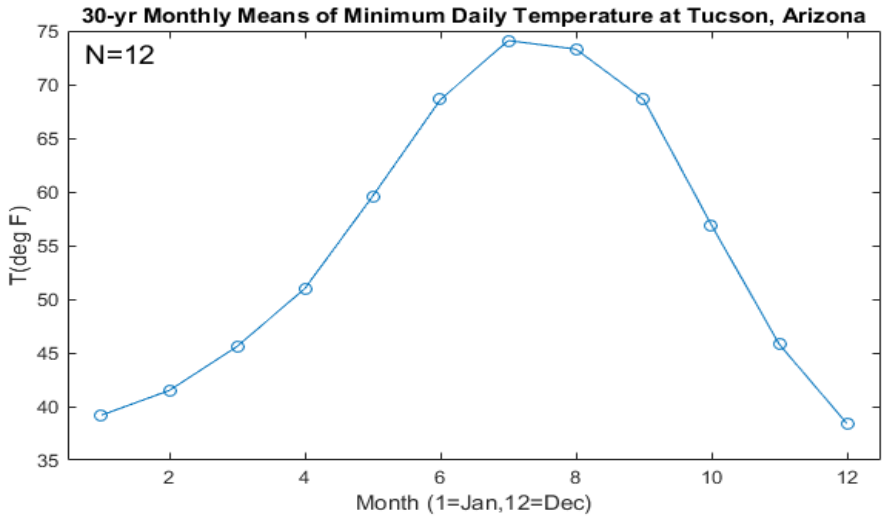

Figure 5. Long-term means of monthly mean daily minimum temperature at Tucson, Arizona. Source: This plot summarizes the annual cycle present in the series plotted in the preceding figure. The plotted data is not, however, itself a time series, as the points do not represent observations at a sequence of times. The plot is a statistical summary of a time series (means computed over the 30 years 1981-2010). Source of data: PRISM Data Explorer (http://www.prism.oregonstate.edu/).

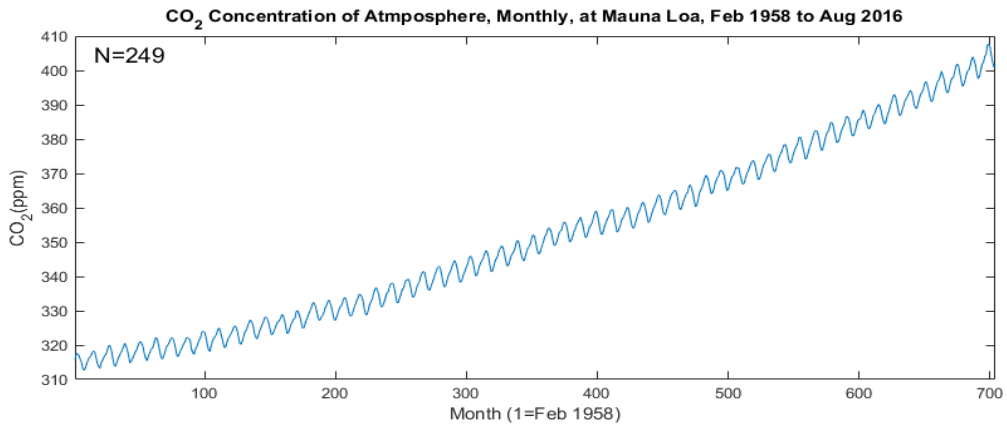

Figure 6. Monthly average atmospheric CO_2 concentration measured at Mauna Loa, March 1958 to November 2016. Series is long (N>>50 observations), continuous-valued, periodic, and nonstationary, with a constant time step of 1 month. An annual cycle related to photosynthesis by vegetation is imposed on a gradual trend in mean due to increasing CO_2 releases from industrialization, etc. Note that the time axis is sequential month, with month 1 corresponding to February, 1958. Source of data: Global Monitoring Division, Earth System Research Laboratory (ftp://aftp.cmdl.noaa.gov/products/trends/co2/co2_mm_mlo.txt).

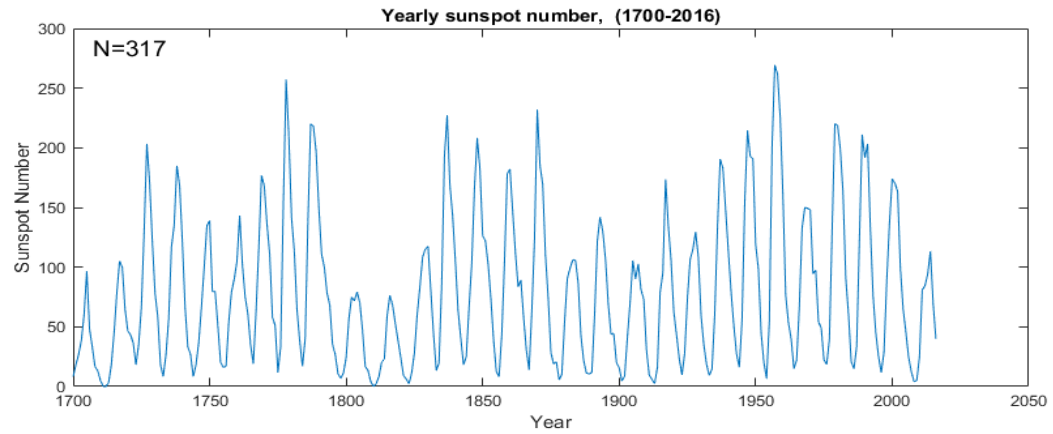

Figure 7. Yearly mean sunspot number, 1700-2016. Series is long (N>>50 observations), continuous-valued and quasi-periodic, with a constant time step of 1 year. The qualifier "quasi" describes the irregularity of the sunspot cycle, whose average period is near 11 years. Moreover, unlike the annual cycle in daily minimum temperature, the physical reason for such variation in solar activity is unclear. The yearly mean total sunspot number is computed as the simple arithmetic mean of the daily total sunspot number over all days of each year. Source of data: Silso, or "Sunspot information and long-term solar observations" (http://www.sidc.be/silso/home).

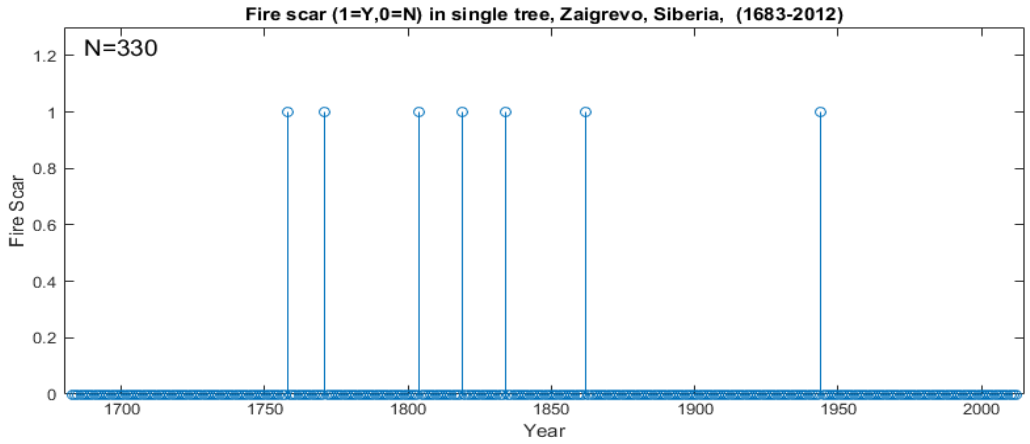

Figure 8. Presence or absence of fire scar in annual tree rings of a single tree in Siberia. Series is long (N>>50 observations), discrete-valued, stationary or not, with a constant time step of 1 year. The series is "discrete-valued" because it can take on only specific values -- 1 or 0 corresponding to fire scar or no fire scar. This series is more appropriately called a "point process" than a time series, with fire scars corresponding to "events." Quantities of interest in the study of such series include the frequency of events in a given time interval and the time between events. Stationarity is impossible to judge from the plot alone. Information on the process might, however, help in this assessment. For example, time-varying emphasis on fire suppression could impose non-stationarity. Source of data: Erica Bigio, personal communication.

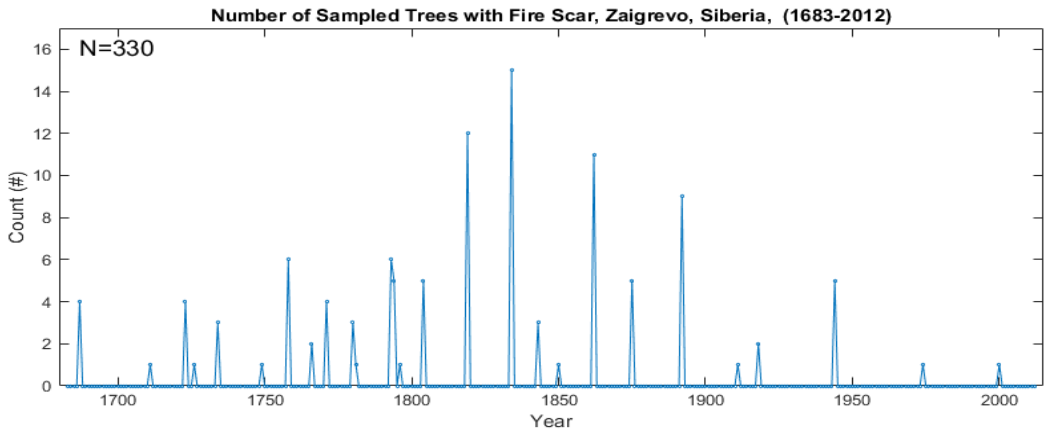

Figure 9. Number of sampled trees fire-scarred each year at tree-ring site in Siberia. Series is long (N>>50 observations), discrete-valued, stationary or not, with a constant time step of 1 year. This discrete-valued series can take on only integer values less than or equal to the number of trees sampled in given year. An obvious possible source of nonstationarity in this series might be time variation in the number of trees sampled. The series is an example of a "count" time series. Gaps of decades occur with zero number of events. Such a series is generally not amenable to study by the methods used in the course. Count series with a large number of possible values and non-zero counts in the vast majority of observations may, however, be suitable. Source of data: Erica Bigio, personal communication.

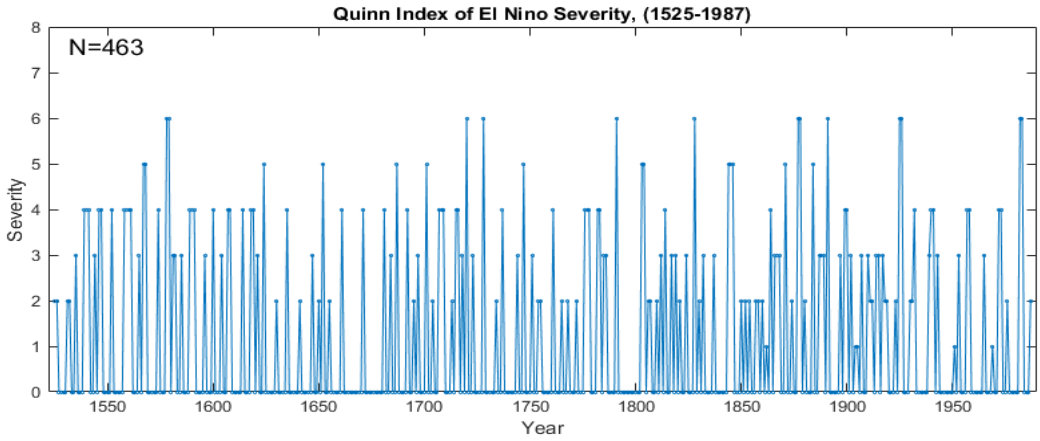

Figure 10. Quinn index of severity of El Nino events, 1525-1987. This is an example of a categorical time series. Series is long (N>>50 observations), discrete-valued, stationary or not, with a constant time step of 1 year. This series is an example of an ordered categorical time series. El Nino is determined to not occur (severity=0) or to occur with severity weak-moderate (1) to very strong (6). Severity is judged by a large number of subjective and objective measures (see reference below). A categorical series generally is not a good candidate for analysis by the methods in the course. The exception is when the categorical variable is "ordered", as this El Nino severity index is, the number of categories is large and the time series is not dominated by a single category (e.g., no El Nino). Source of data: http://research.jisao.washington.edu/data_sets/quinn/).

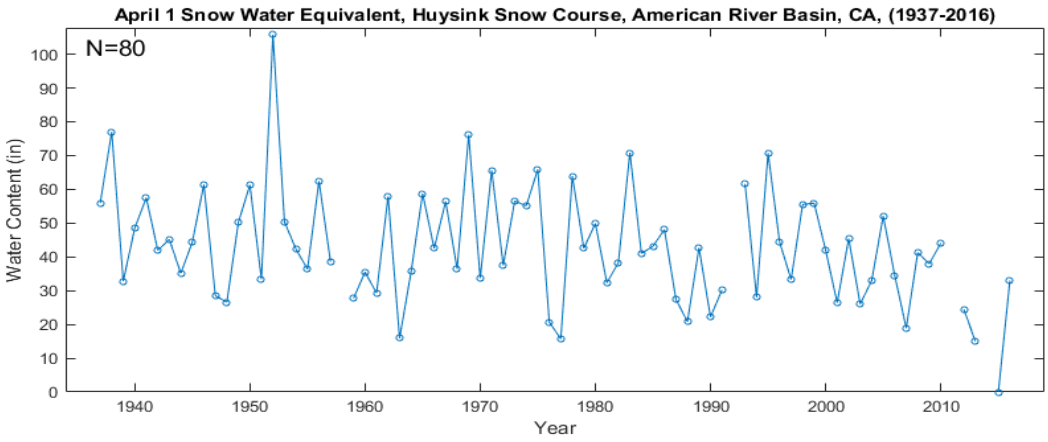

Figure 11. April 1 snow water equivalent at Huysink, California, snow course, 1937-2016. Series is long (N>>50 observations), continuous-valued, and aperiodic, with a constant time step of 1 year and apparent stationarity of mean. Snow water equivalent (SWE) is water content of the snowpack measured at specific times by weighing samples extracted in metal tubes on specific dates. A trait of this series is missing data: no data in 1958, 1992, 2011 and 2014. These are "internal" missing values, in contrast to data that may be missing from leading or trailing years when the series is stored in a multivariate time series matrix. The Matlab scripts for the course require that any internal missing data be estimated beforehand, such the time series to be used is unbroken by missing-data gaps. Source of data: National Science Foundation project on tree-ring signal for snow variables in watershed of the North Fork American River (D. Meko and R. Touchan, PIs).

1.13 Some sample input fles

A simple example using fictional data will serve to illustrate the format and content of the input data for script `geosa1.m`. Consider a V1 data set consisting of 6 tree-ring index chronologies. Assume you have already stored these time series in a tab-delimited data file "`V1Data.txt`". Your paired `txt` metadata file for the V1 series should be named `V1Meta.txt`, and might look like this:

```
V1Data.txt
Year
NaN
MEAF-PSME $ Mesa Alta Fir PSME, standard index     $ Index $ Dimensionless
MEAP-PIST $ Mesa Alta Pine PIST, standard index    $ Index $ Dimensionless
BCWF-PSME $ Bear Can W Fir PSME, standard index    $ Index $ Dimensionless
BCWP-PIST $ Bear Can W Pine PIST, standard index   $ Index $ Dimensionless
FEN-PIPO  $ Fenton Lake PIPO, standard index       $ Index $ Dimensionless
EAU-PSME  $ Echo Amphitheater PSME, standard index $ Index $ Dimensionless
```

The first line tells `geosa1.m` the name of the tab-delimited `txt` file with the time series. The second line says the time step for all time series is a "Year". This time label must exactly match the header in row 1/column 1 of the tab-delimited data file `V1Data.txt`. The third line is "NaN", meaning that any occurrence of "NaN" in the data file should be interpreted as missing data. Remaining lines list the individual V1 time series. For example, the first series has short name "`MEAF-PSME`", long name "`Mesa Alta Fir PSME,standard index`", and is an "`Index`" that is dimensionless – has units "`Dimensionless`".

For this example, files `V1Data.txt` and `V1Meta.txt` apply to the V1 data. You also need to prepare corresponding pairs of files for the V2 and V3 data.

1.14 Running geosa1.m

`Geosa1` is a function that reads the ascii `txt` files of time series and metadata and stores the data in a `mat` file that is used in the rest of the course. `Geosa1` will prompt you for your last name, and will give the `mat` file this name (e.g., `Lincoln.mat`). `Geosa1` stores all the data in this single `mat` file. Before running `geosa1`, make sure that the required input data files are in the current Matlab working directory (see `appendixa.pdf`). If you run into problems, detailed instructions for running `geosa1` are found elsewhere (see "**Running goesa1.m"** in `a1.pdf`).

1.15 Checking that the structure variable has been created and stored

After running `geosa1`, you should have a `mat` file (e.g., `Lincoln.mat`) in your current working directory, and that `mat` file should hold vlist (a list of variables in the `mat` file), and a subset of time series structure variables V1, V2, V3. Say the `mat` file is named `Lincoln.mat`. Check out contents of the `mat` file by typing commands:

`>> what`

Shows a list of `mat` files in the command window. You should see `Lincoln.mat` in the list

```
>>clear
>>load Lincoln.mat
```
clears the workspace and loads the .mat file you have created.

```
>> whos
```
lists variables in the workspace. You should see vlist and one or more of V1, V2, V3 there now.

```
>>vlist
```
lists the contents of variable vlist

```
>>V1
```
Typing a variable's name displays the contents of the variable in the command window. Names are CASE SENSITIVE. When you type V1, you see a list of all the fields in the structure variable V1.

```
>>V1.id
```
Type the structure followed by a period and a fieldname and you get the expanded contents of the field. So, for example, lists the short series names, or ids

```
>>V1.name
```
lists the names long series names

```
>>V1.tsm
```
lists the time series themselves.

Note that V1.tsm has only the time series data, not the vector of years. V1.time is the year vector. You can generate a quick time series plot of the first and third series, with a legend of ids, by entering this sequence of commands

```
>>t=V1.time;
>>x1=V1.tsm(:,1);
>>x3=V1.tsm(:,3);
>>plot(t,x1,'-',t,x3,'-');
>>ylabel(V1.label{1});
>>xlabel(V1.increment);
>>legend(V1.id{1},V1.id{3});
```

The first three commands rename variables so that the plot syntax is less cluttered. The next commands plot the series and label the plot. Entering the commands one by one at the command prompt is one way to use Matlab. A better way is to organize the commands in an ascii file called a "script" and then run the script from the command prompt. Using scripts avoids having to re-enter commands each time your want to repeat an analysis or modify some part of the analysis or the plots. The interactive tools at the top of the figure window are another way to change attributes of plots. Please refer to the Matlab help files for more information on graphics.

Matlab functions, like scripts, are also ascii files containing commands, but are more general in that they may be applied to different problems by changing the input arguments. The `plot` command above is a function with input arguments the x-data, y data and line-type for the plot. Matlab has many built-in functions for data analysis, and you can extend this capability with your own functions or user-written functions available over the Web.

It is helpful to learn some basic elements of Matlab programming if you want to apply Matlab functions other than those covered in the course, or if you want to tailor your own data analysis. But you do not need to program in Matlab for this course. The script files and functions needed for all analyses have already been written and are provided for the course.

1.16 Turning in assignments as figures and captions

Assignments for the course consist of running Matlab scripts or functions demonstrating the time series methods and writing up brief interpretations of the results. Each assignment should be turned in as a `pdf` document with figures and captions only. The figures are generated by the Matlab scripts. The captions are your answers to the questions in the assignment. Please refer to file `appendixb.pdf` ("Appendix B -- Submitting Assignments") for instructions on preparing and submitting assignments.

1.17 References

Brillinger, D. R. (1994), Time series, point processes, and hybrids, Can. J. Stat., 22 (2), 177–206.

Chatfield, C. (2004), The Analysis of Time Series: An Introduction, sixth ed., Chapman & Hall/CRC, New York, 327 pp.