

## 10.2: R and time

If we measure same object multiple times, especially at regular (sampling) intervals, we will finally have the *time series*, specific type of measurement data. While many common options of data analysis are applicable to time series, there are multiple specific methods and plots.

Time series frequently have two components, non-random and *random*. The first could in turn contain the *seasonal* component which is related with time periodically, like year seasons or day and night. The *trend* is the second part of non-random component, it is both no-random and non-periodical.

If time series has the non-random component, the later values should correlate with earlier values. This is *autocorrelation*. Autocorrelation has lags, intervals of time where correlation is maximal. These lags could be organized hierarchically.

Different time series could be *cross-correlated* if they are related.

If the goal is to analyze the time series and (1) fill the gaps within (*interpolation*) or (2) make forecast (*extrapolation*), then one need to create the time series *model* (for example, with `arima()` function).

But before the start, one will need to convert the ordinary data frame or vector into time series. Conversion of dates is probably most complicated:

In that example, we showed how to use `as.Date()` function to convert one type to another. Actually, our recommendation is to use the fully numerical date:

The advantage of this system is that dates here are accessible (for example, for sorting) both as numbers and as dates.

And here is how to create time series of the regular type:

(If the time series is irregular, one may want to apply `its()` from the `its` package.)

It is possible to convert the whole matrix. In that case, every column will become the time series:

Generic `plot()` function “knows” how to show the time series (Figure 10.2.1:

(There is also specialized `ts.plot()` function.)

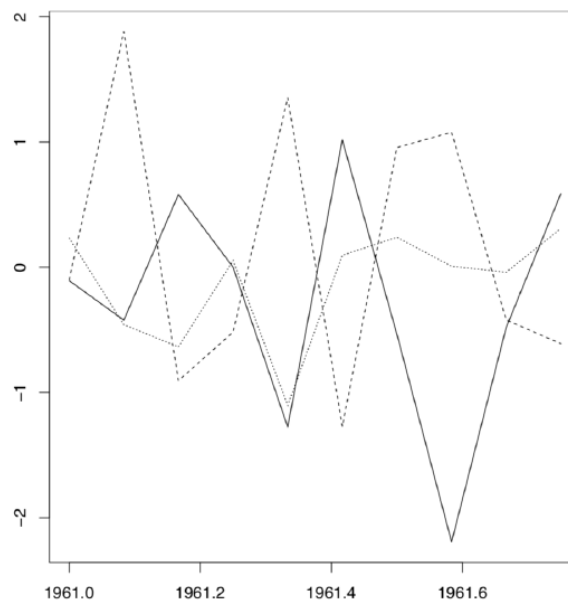


Figure 10.2.1 Three time series with a common time.

There are numerous analytical methods applicable to time series. We will show some of them on the example of “non-stop” observations on carnivorous plant—sundew (*Drosera rotundifolia*). In nature, leaves of sundew are constantly open and close in hope to catch and then digest the insect prey (Figure 10.2.2). File `sundew.txt` contains results of observations related with the fourth leaf of the second plant in the group observed. The leaf condition was noted every 40 minutes, and there were 36 observations per 24 hours. We will try to make the time series from `SHAPE` column which encodes the shape of leaf blade (1 flat, 2 concave), it is the ranked data since it is possible to imagine the `SHAPE` = 1.5. Command `file.show()` reveals this structure:

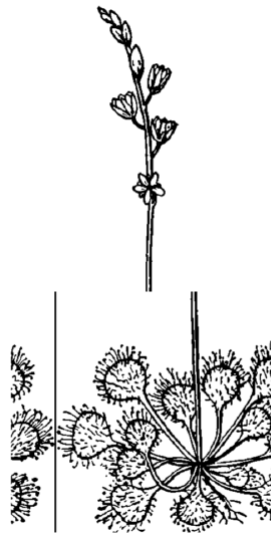


Figure 10.2.2 Sundew, *Drosera rotundifolia*. These carnivorous plants know their time to eat.

```
WET;SHAPE
2;1
1;1
1;1
...
```

Now we can read the file and check it:

Everything looks fine, there are no visible errors or outliers. Now convert the **SHAPE** variable into time series:

Let us check it:

Looks perfect because our observations lasted for slightly more than 3 days. Now access the periodicity of the time series (seasonal component) and check out the possible trend (Figure 10.2.3):

(Please note also how **expression()** was used to make part of the title italic, like it is traditional in biology.)

Command **acf()** (auto-correlation function) outputs coefficients of autocorrelation and also draws the autocorrelation plot. In our case, significant periodicity is absent because almost all pikes lay within the confidence interval. Only first tree pikes are outside, these correspond with lags lower than 0.05 day (about 1 hour or less). It means that within one hour, the leaf shape will stay the same. On larger intervals (we have 24 h period), these predictions are not quite possible.

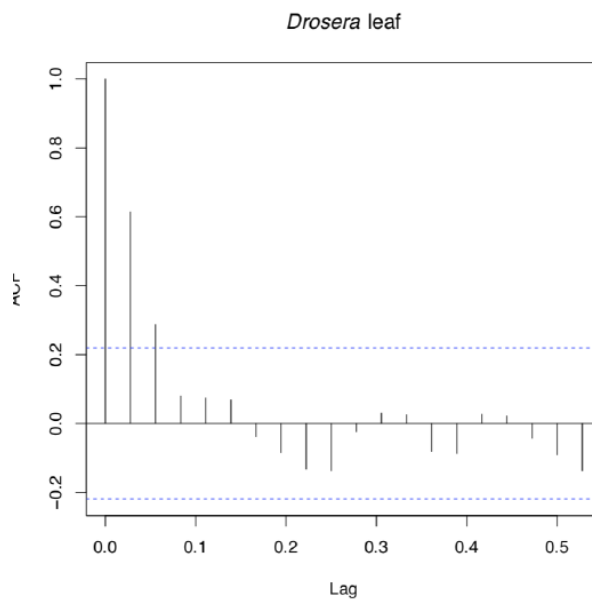


Figure 10.2.3 Autocorrelation plot for the sundew leaf.

However, there is a tendency in pikes: they are much smaller to the right. It could be the sign of trend. Check it (Figure 10.2.4):

WET is the second character in our sundew dataset. It shows the wetness of the leaf. Does wetness have the same periodicity and trend as the leaf shape?

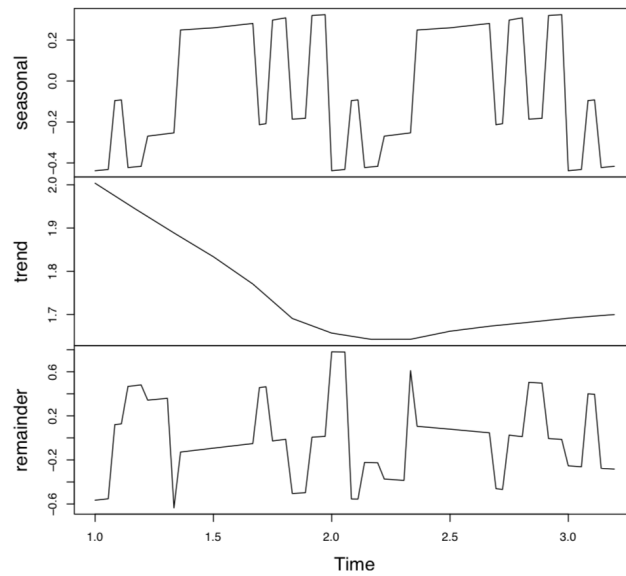


Figure 10.2.4 Seasonal decomposition plot for the leaf of sundew. The possible trend is shown in the middle.

This page titled 10.2: R and time is shared under a [Public Domain](#) license and was authored, remixed, and/or curated by Alexey Shipunov via source content that was edited to the style and standards of the LibreTexts platform.