Time series representation in R

Usue Mori

This tutorial will be divided into two parts. The first one will analyze the packages in R that can be used to represent time series of different types. In the second part, we will learn about some popular time series representation methods.

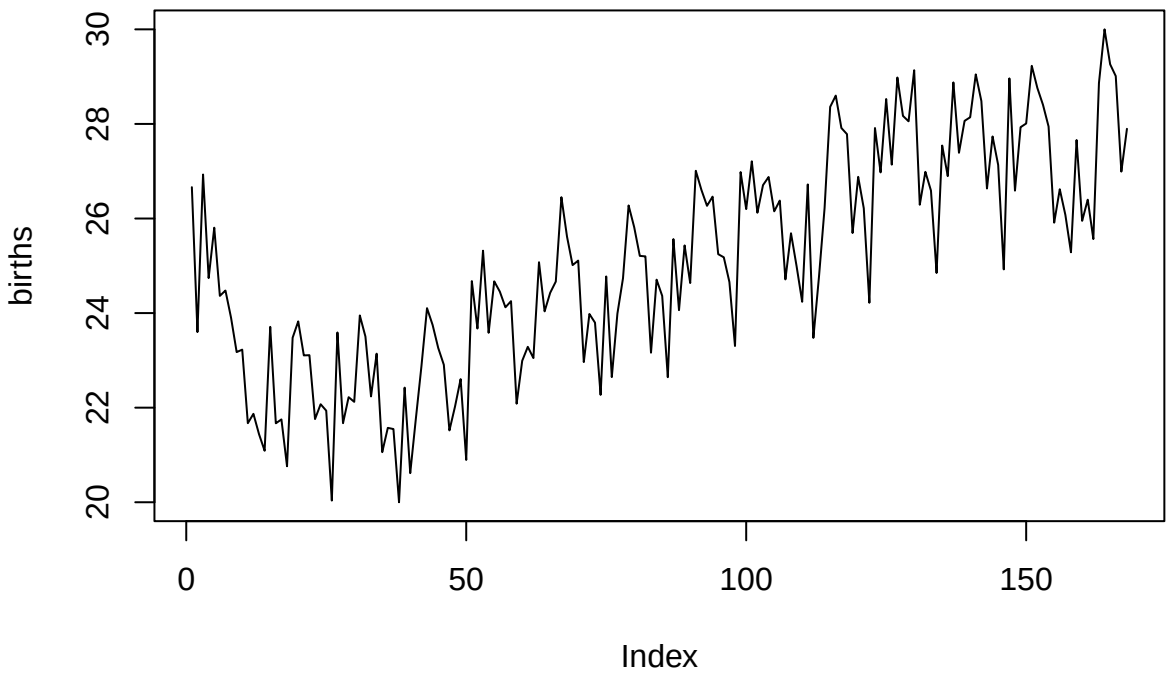
Time series objects in R

Time series can be defined in many different ways in R. The first and most simple representation is by using a vector:

```
births <- scan("http://robjhyndman.com/tsdldata/data/nybirths.dat")
head(births)
```

```
[1] 26.663 23.598 26.931 24.740 25.806 24.364
```

```
plot(births, type="l")
```



This vector collects the number of births in New York city from January 1946 to December 1959.This type of object will only be useful when our time series are regularly sampled (i.e. the time between measurements remains constant) and when we do not care about the specific timestamps, but only the values of the time series.

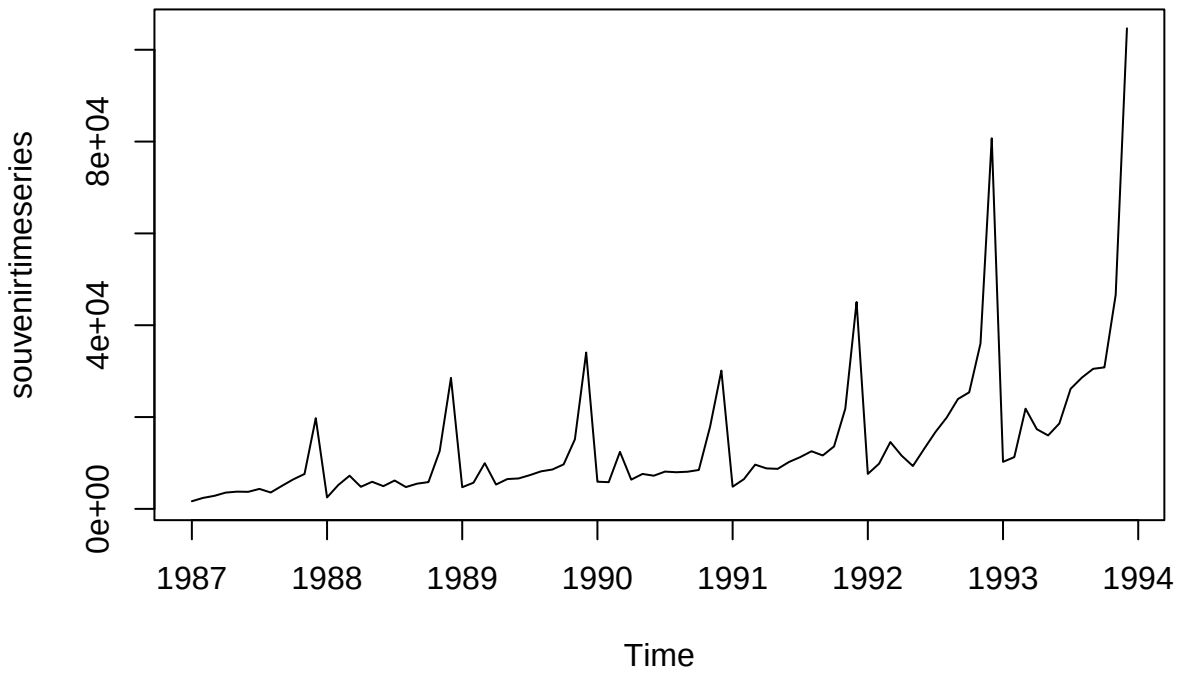
In view of this, in several packages of R, we can find specific objects which can be used to represent more complex types of time series. These are two of them:

- `ts`: it is part of the basic package of R and is the basic class for regularly spaced time series using numeric time stamps. To begin with, it can be used for univariate time series as follows:

```
souvenir <- scan("http://robjhyndman.com/tsdldata/data/fancy.dat")
souvenirtimeseries <- ts(souvenir, frequency=12, start=c(1987,1))
souvenirtimeseries
```

```
          Jan       Feb       Mar       Apr       May       Jun       Jul
1987   1664.81   2397.53   2840.71   3547.29   3752.96   3714.74   4349.61
1988   2499.81   5198.24   7225.14   4806.03   5900.88   4951.34   6179.12
1989   4717.02   5702.63   9957.58   5304.78   6492.43   6630.80   7349.62
1990   5921.10   5814.58  12421.25   6369.77   7609.12   7224.75   8121.22
1991   4826.64   6470.23   9638.77   8821.17   8722.37  10209.48  11276.55
1992   7615.03   9849.69  14558.40  11587.33   9332.56  13082.09  16732.78
1993  10243.24  11266.88  21826.84  17357.33  15997.79  18601.53  26155.15
          Aug       Sep       Oct       Nov       Dec
1987   3566.34   5021.82   6423.48   7600.60  19756.21
1988   4752.15   5496.43   5835.10  12600.08  28541.72
1989   8176.62   8573.17   9690.50  15151.84  34061.01
1990   7979.25   8093.06   8476.70  17914.66  30114.41
1991  12552.22  11637.39  13606.89  21822.11  45060.69
1992  19888.61  23933.38  25391.35  36024.80  80721.71
1993  28586.52  30505.41  30821.33  46634.38 104660.67
```

```
plot.ts(souvenirtimeseries)
```



This time series records the monthly sales for a souvenir shop at a beach resort town in Queensland, Australia, for January 1987-December 1993. As can be seen, when using the `ts` class, in addition to the actual values of the series, more information is saved (frequency, timestamps, etc.).

- `zoo`: contained in the `zoo` package, it provides infrastructure for regularly and irregularly spaced time series and allows for multiple formats in the definition of the timestamps.

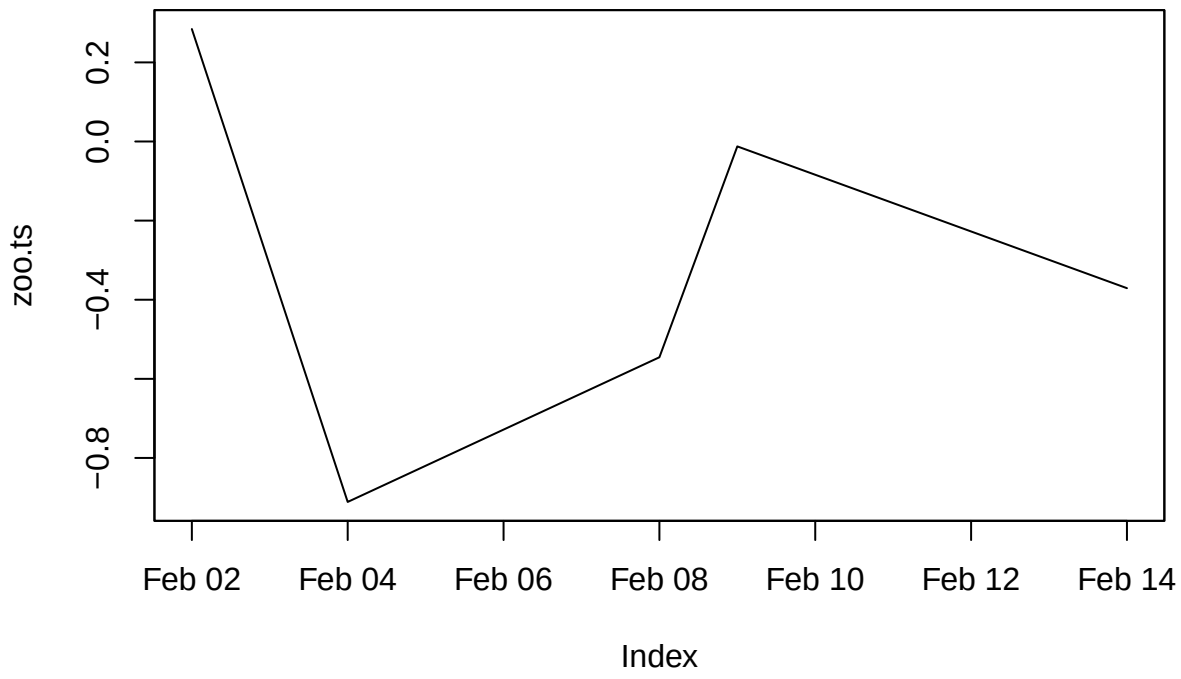
```
library(zoo)
```

```
Attaching package: 'zoo'

The following objects are masked from 'package:base':

    as.Date, as.Date.numeric
```

```
zoo.Date <- as.Date("2003-02-01") + c(1, 3, 7, 8, 13)
zoo.ts <- zoo(rnorm(5), zoo.Date)
zoo.ts
```

```
 2003-02-02  2003-02-04  2003-02-08  2003-02-09  2003-02-14
 0.28429470 -0.91118110 -0.54547198 -0.01238119 -0.37078620
```

```r
plot(zoo.ts)
```



The type of object to use to represent our time series will depend on the type of data that we are dealing with but also the requirements of other functions that we are going to use to analyze or mine the series. For more information on the methods associated to the `ts` and `zoo` classes, access the R help.

To finish, we must say that in addition to these packages, there is a Task View of R specifically dedicated to time series: https://cran.r-project.org/web/views/TimeSeries.html. Here you can find a summary of all the functionalities and packages available for time series analysis in R.

Time series representation methods

As mentioned in class, time series can be represented and analyzed in their raw form. However, in many occasions, due to the high dimensionality, noise, or other factors, other representation methods can also be interesting.

In this section we will learn to represent time series using different representation methods implemented in the `TSrepr` package in R. Note that the objective of all these representation methods is to:

- Obtain a reduction of the time series dimensionality
- Retain the (essential) shape characteristics of the time series.
- Handle or reduce the noise present in this type of data.

First of all, let's install and load the package:

```
install.packages("TSrepr")
```

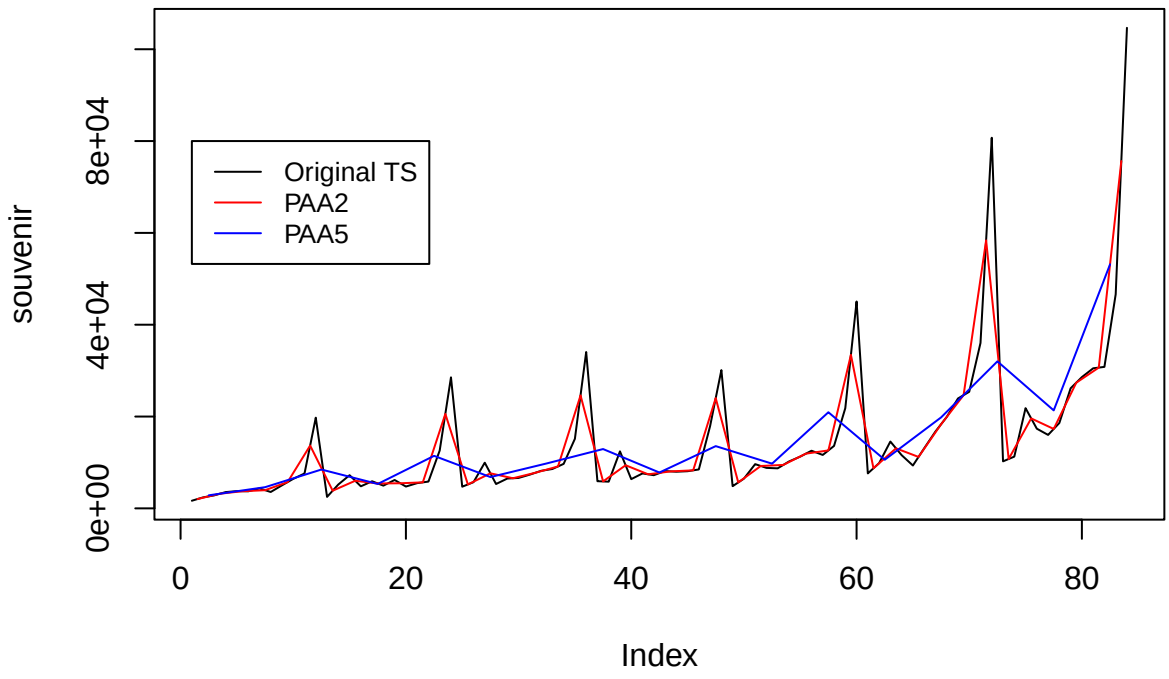
```
Installing package into '/usr/local/lib/R/site-library'
(as 'lib' is unspecified)
```

```
library(TSrepr)
```

Non data-adaptive methods

In these types of methods, the parameters of transformation must be fixed by the user and are not automatically adapted to the data at hand. One of the most popular non data-adaptive methods is PAA (Piecewise Aggregate Approximation). This method, divides the time series into subsequences of fixed size q and substitutes each subsequence by a summary value of the observations within, typically the mean. The parameter q must be chosen by the user. Let's apply this method to the `souvernir` time series introduced in previous sections:

```
souvenirs_paa2 <- repr_paa(souvenirtimeseries, q = 2, func = mean)
souvenirs_paa5 <- repr_paa(souvenirtimeseries, q = 5, func = mean)
plot(souvenir, type="l")
lines(seq(1.5, length(souvenir), by=2), souvenirs_paa2, type="l", col="red")
lines(seq(2.5, length(souvenir), by=5), souvenirs_paa5, type="l", col="blue")
legend(1, 80000, legend=c("Original TS", "PAA2", "PAA5"),
       col=c("black", "red", "blue"), lty=1, cex=0.8)
```

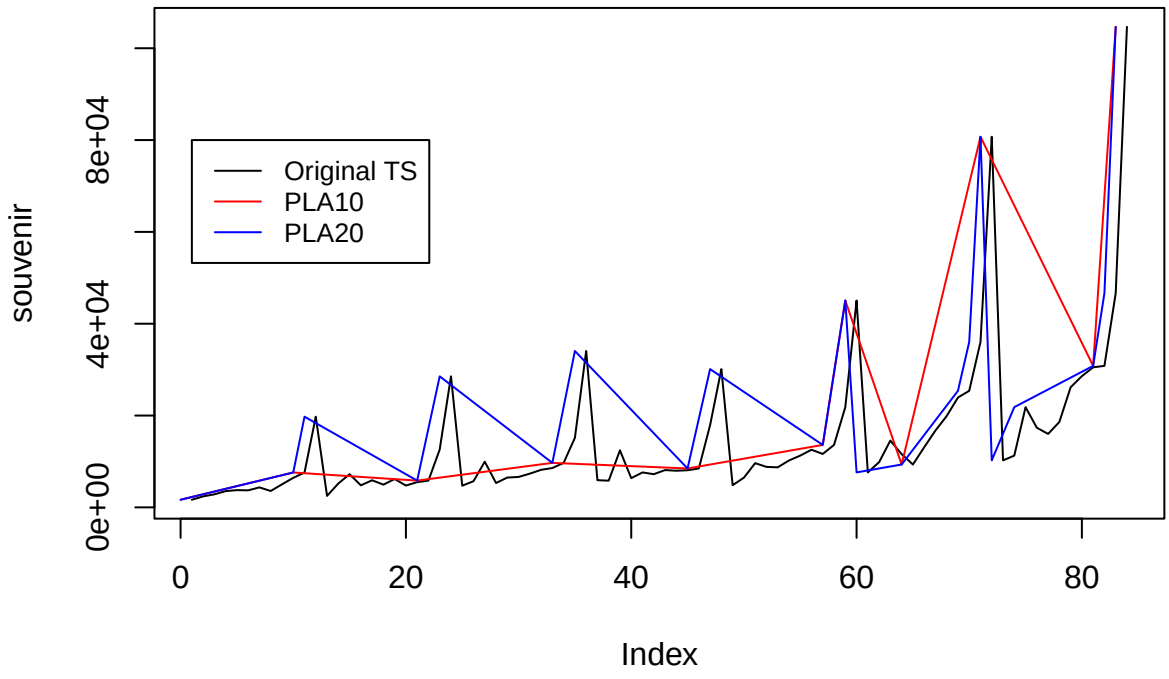


As can be seen, for smaller q values, the representation is closer to the original series.

Data adaptive methods

In this case, the parameters of transformation are learnt within the method and depend on the data at hand. One of the most important data-adaptive methods is the Piecewise Linear Approximation (PLA). With this method, the time series is divided into subsequences and each subsequence is approximated by a straight line. The division of the time series into subsequences is done differently for each time series and depends on its characteristics. There are many method to do this division [**?**], and in the implementation in package `TSrepr` it is done based on feature points, that is, points where abrupt changes happen in the time series.

```
souvenirs_pla10 <- repr_pla(souvenirtimeseries, times=10, return="both")
souvenirs_pla20 <- repr_pla(souvenirtimeseries, times=20, return="both")
plot(souvenir, type="l")
lines(souvenirs_pla10$places, souvenirs_pla10$points, type="l",, col="red")
lines(souvenirs_pla20$places, souvenirs_pla20$points, type="l",, col="blue")
legend(1, 80000, legend=c("Original TS", "PLA10", "PLA20"),
       col=c("black", "red", "blue"), lty=1, cex=0.8)
```



In this cases, the larger the parameter `times` the closer to the original series the representation becomes.

Exercises

- Find out how we can use the classes `ts` and `zoo` to save multivariate time series. Generate a synthetic semestral multivariate time series of 5 variables and save it in an object of type `ts`. Generate an irregularly sampled multivariate time series using `xts`.

- Find out about other popular non data-adaptive representation methods such as the DFT (Discrete Fourier Transform) implemented in the `TSrepr` package. How do these methods work? Why are they non-data adaptive? Try to apply them to the `souvernir` time series and analyze the results by using different parameter settings.
- Find out about SAX, a very popular data-adaptive representation method implemented in the `TSrepr` package. How does this method work? Why is it considered data adaptive? Try to apply it to the `souvernir` time series and analyze the results when using different parameter settings.

References

- Esling, P., and Agon C.. 2012. "Time-series data mining." ACM Computing Surveys 45 (1): 1–34.
- Laurinec P. TSrepr vignette. https://cran.r-project.org/web/packages/TSrepr/vignettes/TSrepr_representations_of_time_series.html
- Lin, J., Keogh, E., Wei, L., & Lonardi, S. (2007). Experiencing SAX: a novel symbolic representation of time series. Data Mining and knowledge discovery, 15(2), 107-144.