

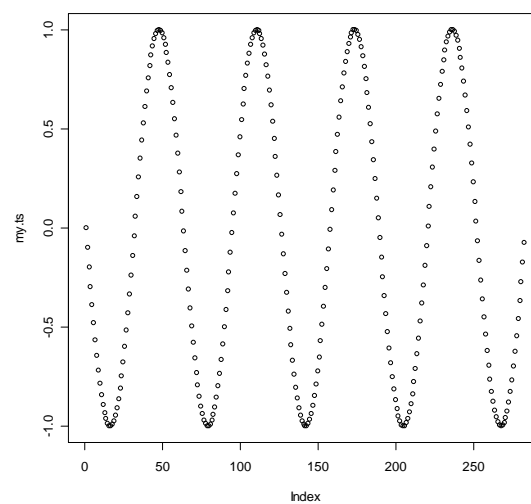
A short introduction to time series analysis in R

The key point in time series analysis is that observations tend to show serial temporal autocorrelation that needs to be accounted for in statistical analyses.

Example 1

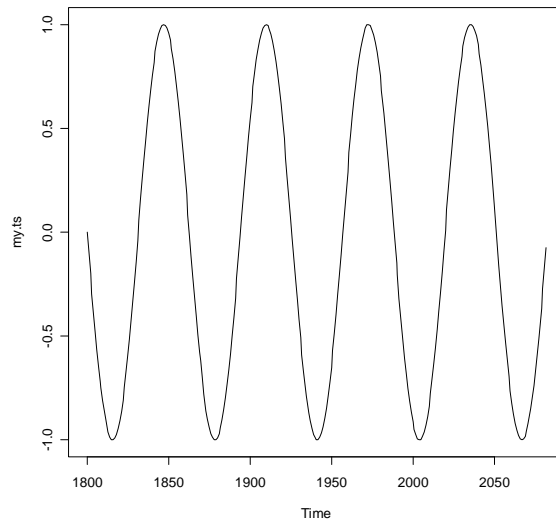
For example, let's create a time series of points ranging from π to $n\pi$ in steps of 0.1:

```
my.ts=sin(seq(pi,10*pi,0.1))
```



To tell R that these data actually represent a **time series**, we need to re-define the sequence to be a time series; let us assume that the time series starts in the year 1800:

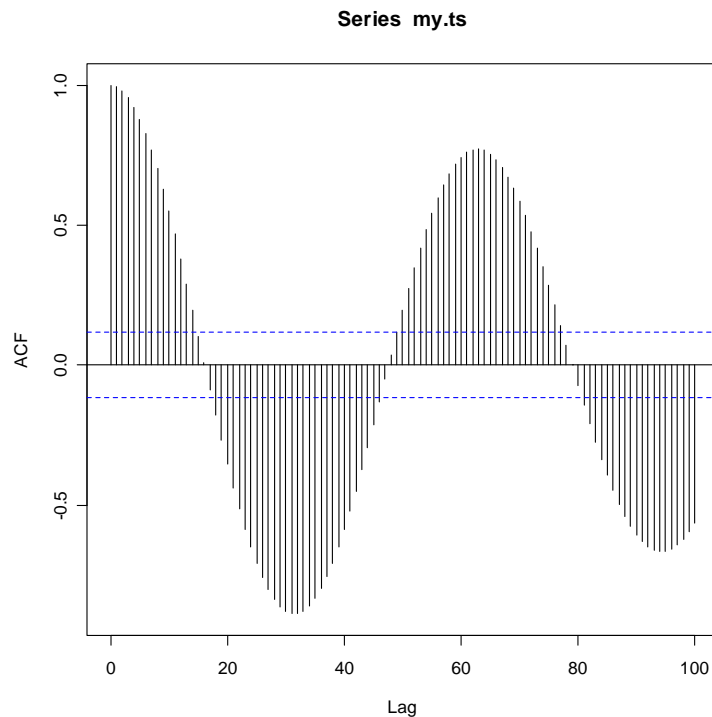
```
my.ts=ts(my.ts,start=1800)  
plot(my.ts)
```



Now we know *exactly* that these data points are perfectly autocorrelated; this means, for each point n it is known that the next $(n+k)$ points will follow a sine function.

Let us see what R's built-in autocorrelation function tells us:

```
acf(my.ts, lag.max=100)
```



What we can see from this plot is that the underlying data-generating sine function is correctly represented in the autocorrelation function. We do, however, see a slight decline in the acf with increasing lag due to small prediction errors. However, this autocorrelation is highly significant (indicated by all vertical lines crossing the dashed horizontal ones).

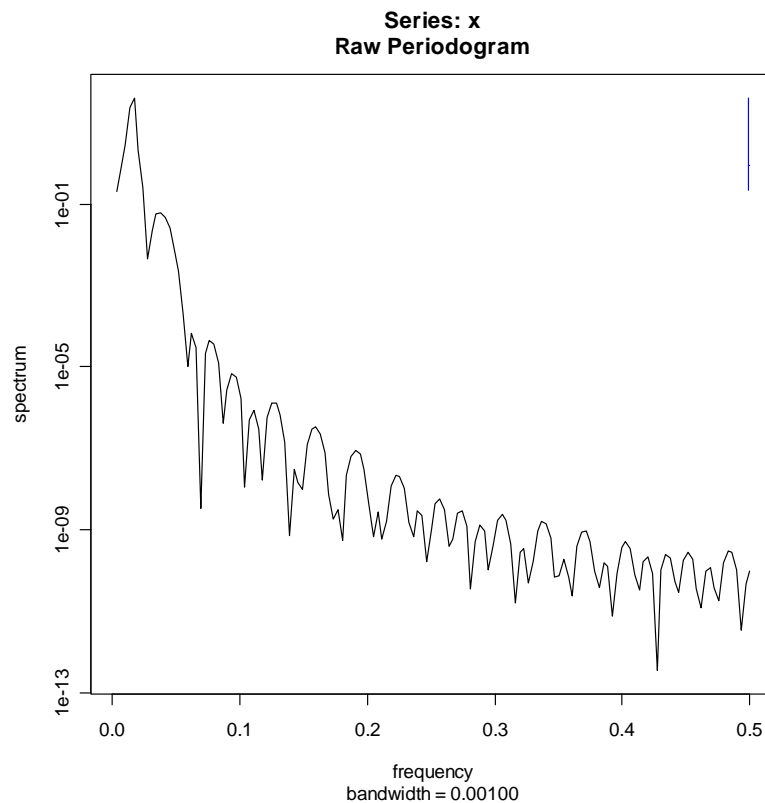
What about the numerical values of this autocorrelation function?

```
acf(my.ts,lag.max=100,plot=F)
```

```
>Autocorrelations of series 'my.ts', by lag
```

0	1	2	3	4	5	6	7	8	9	10
1.000	0.995	0.980	0.955	0.922	0.879	0.828	0.768	0.702	0.629	0.551
11	12	13	14	15	16	17	18	19	20	21
0.467	0.380	0.289	0.196	0.102	0.007	-0.086	-0.179	-0.268	-0.355	-0.437
22	23	24	25	26	27	28	29	30	31	32
-0.514	-0.586	-0.651	-0.709	-0.760	-0.803	-0.837	-0.863	-0.880	-0.888	-0.887
33	34	35	36	37	38	39	40	41	42	43
-0.878	-0.860	-0.833	-0.798	-0.756	-0.707	-0.651	-0.589	-0.521	-0.449	-0.373
44	45	46	47	48	49	50	51	52	53	54
-0.294	-0.213	-0.131	-0.048	0.035	0.117	0.197	0.274	0.348	0.418	0.483
55	56	57	58	59	60	61	62	63	64	65
0.542	0.596	0.643	0.683	0.717	0.742	0.760	0.771	0.773	0.768	0.755
66	67	68	69	70	71	72	73	74	75	76
0.735	0.708	0.673	0.633	0.586	0.534	0.478	0.417	0.352	0.284	0.215
77	78	79	80	81	82	83	84	85	86	87
0.144	0.072	0.000	-0.072	-0.141	-0.209	-0.274	-0.336	-0.394	-0.447	-0.496
88	89	90	91	92	93	94	95	96	97	98
-0.538	-0.576	-0.607	-0.632	-0.650	-0.661	-0.666	-0.665	-0.657	-0.642	-0.621
99	100									
-0.595	-0.562									

The spectrum() function provides a spectral decomposition of our time series:



The spectrum has a maximum value at about 0.03 (which is on a log scale), for which also a confidence estimate is given (the vertical bar to the right).

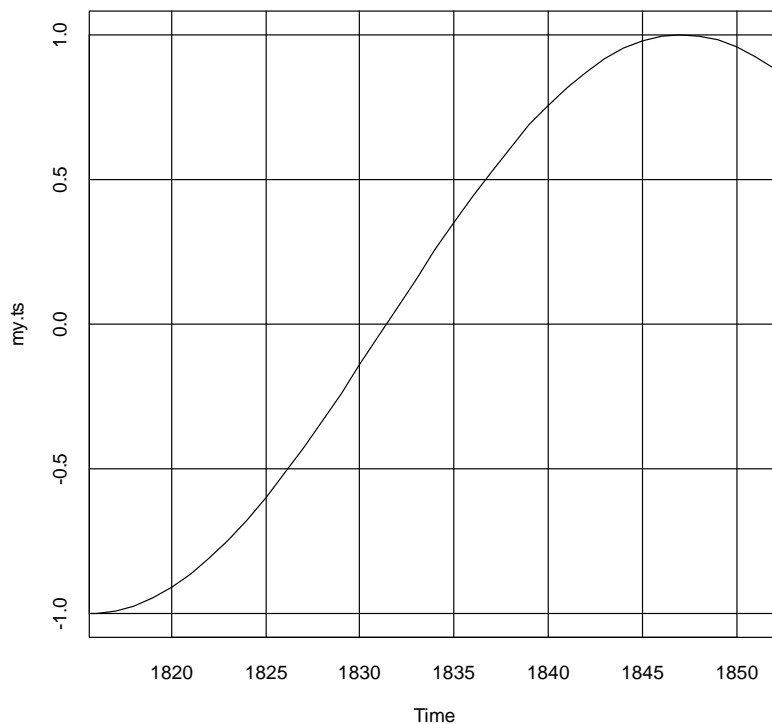
In summary, the periodogram shows that the cycles have a period of about $1/0.03=33$ years. Let us check if this approximation holds:

We know from the **autocorrelation function** that there is a local minimum in the year 1816 (roughly 0.0); the value at 1831 is almost perfectly negatively correlated with the one from 1800, and the cycle approaches zero again in 1848. The next maximum is in 1863, and so on. Thus, our autocorrelation function predicts a period of about $(1863-1831)=32$ years.

This is well in accordance with our above guess of 33 years.

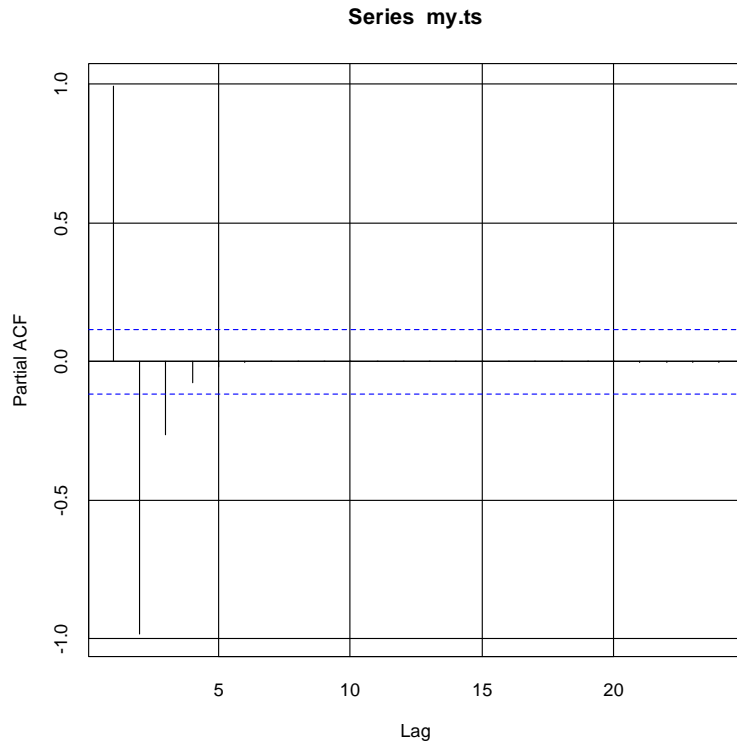
An inspection of the first few values from the 1817 onward gives:

```
par(tck=1)
plot(my.ts,type="l",xlim=c(1817,1851))
```



Let us now inspect the **partial** autocorrelation function:

```
plot(acf(my.ts,type="p"))
```



```
acf(my.ts,type="p",plot=F)
```

```
>Partial autocorrelations of series 'my.ts', by lag
```

```

  1      2      3      4      5      6      7      8      9     10     11
0.995 -0.985 -0.266 -0.075 -0.020 -0.004  0.001  0.002  0.003  0.003  0.002
 12     13     14     15     16     17     18     19     20     21     22
0.002  0.002  0.001  0.001  0.001  0.000  0.000 -0.001 -0.001 -0.001 -0.002
 23     24
-0.002 -0.002
```

We see that at lags 2 and 3 the partial acf is significant, but after that nothing more goes on.

One important thing to remember here is that it may be difficult to infer about mechanisms by looking at the dynamics of a system. Thus, while we can well inspect the dynamical behavior of our time series cycles, mathematical tools would be needed to detect the underlying function that has generated the time series.

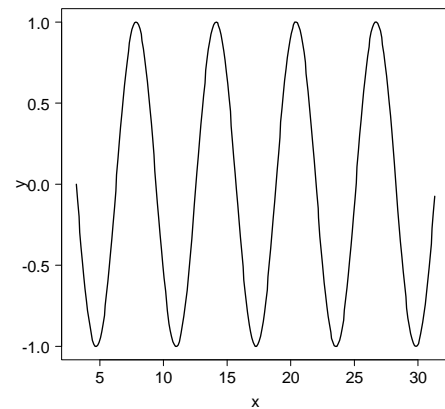
Example 2

Now let us examine a time series that has a linear trend. We can create the trend artificially like this:

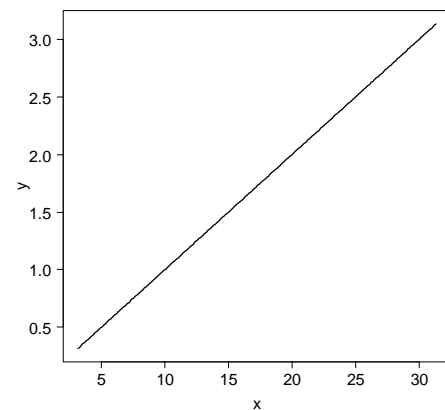
```
x=seq(pi,10*pi,0.1)
my.ts=0.1*x+sin(x)
```

We thus have a sine function to which a linear function has been added. The two components are:

a) $y=\sin(x)$

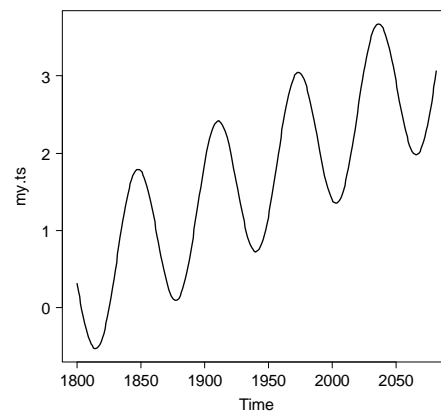


b) $y=0.1*x$



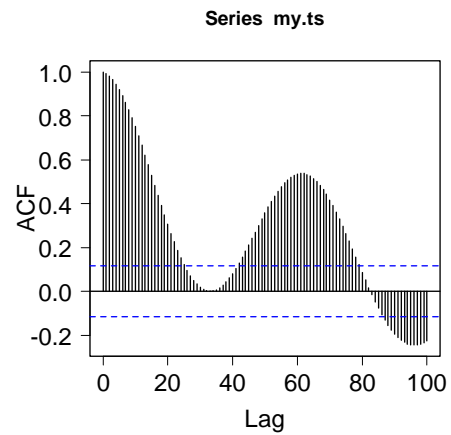
So let us now inspect the combined "pseudo time series":

```
my.ts=0.1*x+sin(x)  
my.ts=ts(my.ts,start=1800)  
plot(my.ts)
```



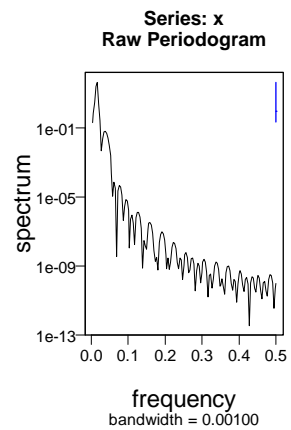
First of all, we can again check for autocorrelation in this time series:

```
plot(acf(my.ts,lag.max=100))
```



Further, we could look at a spectral decomposition of the time series:

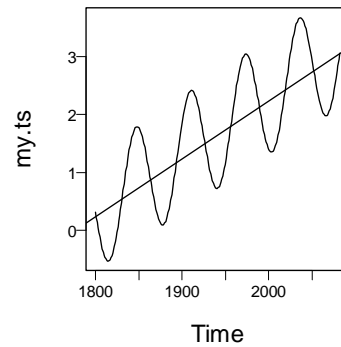
```
spectrum(my.ts)
```



Now let us test for a linear trend in this time series (of which we know it is there):

```
modell1=lm(my.ts~I(1801:2083))  
summary(modell1)
```

```
plot(my.ts)
abline(modell)
```



Finally, the `summary(modell)` command gives:

Call:

```
lm(formula = my.ts ~ I(1801:2083))
```

Residuals:

Min	1Q	Median	3Q	Max
-0.9295	-0.6860	-0.0790	0.6985	1.0706

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-1.776e+01	9.981e-01	-17.79	<2e-16 ***
I(1801:2083)	9.995e-03	5.135e-04	19.46	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

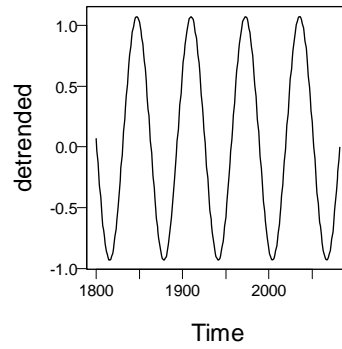
Residual standard error: 0.7057 on 281 degrees of freedom
Multiple R-Squared: 0.5741, Adjusted R-squared: 0.5726
F-statistic: 378.9 on 1 and 281 DF, p-value: < 2.2e-16

We conclude that the model has an intercept of -1.77 and a slope of 0.01. Note that these values are for the time series starting at 1800.

Now, finally, let us de-trend the time series:

```
detrended=my.ts-  
predict(model1)
```

```
plot(detrended)
```



Which leaves us back with the two components we have used to create the time series initially.