Sunspot activity: analyzing and visualizing 150 years of time series data

From

Essentials of Programming in Mathematica

Exercises from Chapter 4, Patterns and rules

© 2015 Paul Wellin

4.3 Examples

Sunspot activity

Sunspots are dark spots on the sun where the surface temperature is reduced by the effects of strong magnetic activity. They were first observed and recorded by the Chinese over 2000 years ago. In the year 1610, soon after the invention of the telescope, Galileo (1564–1642) spotted them, thus dealing one of several blows to religious dogma of the time which insisted that the heavens and stars were perfect objects.

Extraordinary levels of ultraviolet and x-ray emissions from increased solar activity have pronounced effects on the Earth's upper atmosphere and so detecting patterns in the frequency of sunspot activity is an important aspect of current solar research. Fortunately, data is available going back to 1818. The time-series data that we will import in this example requires some extraction, cleaning, and filtering before we can analyze and visualize it, typical tasks when working with real data. Transformation rules provide a clean and efficient mechanism to perform these tasks.

Start by importing the data from the WDC-SILSO, Royal Observatory of Belgium.

```
In[1]:= data = Import["http://sidc.oma.be/silso/DATA/SN_d_tot_V2.0.txt", "Table"];
```

There are over 70 000 observations (or records).

```
In[2]:= Length[data]

Out[2]= 72 226
```

Our task here is to filter all of the data and make a plot of the sunspot numbers over the time period covered by the data set. We start by converting the data to a form that can be used by `TimeSeries` and `DateListPlot`; that is, convert the data to a list of the form {{*date*, *value*}, …}. We will first prototype with a small number of data points and, when we are satisfied with the code, apply it to the entire data set.

Here are the first two and last two records or data points from the imported data.

```
In[3]:= test = Part[data, {1, 2, -2, -1}]

Out[3]= {{1818, 1, 1, 1818., -1, -1., 0}, {1818, 1, 2, 1818., -1, -1., 0},
         {2015, 9, 29, 2015.74, 117, 8.8, 29, *}, {2015, 9, 30, 2015.75, 95, 6.3, 31, *}}
```

Next we need to put the data into a form suitable for the time series functions. Each observation is of the form {*y*, *m*, *d*, *yf*, *sn*, *sd*, *obs*, *ind*}, where *y* is the year, *m* is the month, *d* is the day, *yf* is the date in fraction of a year, *sn* is the sunspot number, *sd* is the standard deviation of the sunspot numbers, *obs* is the raw number of observations used to compute *sn*, and *ind* is a value used to indicate if the sunspot number value is still provisional (see the SILSO website for details).

Notice that earlier data points have seven elements in each sublist while later data consists of eight-element lists. The following rule will remove the indicator element if present and leave seven-element lists unchanged.

```
In[4]:= test2 = test /. lis : {y_, m_, d_, _, _, _, _, _} :> Most[lis]

Out[4]= {{1818, 1, 1, 1818., -1, -1., 0}, {1818, 1, 2, 1818., -1, -1., 0},
         {2015, 9, 29, 2015.74, 117, 8.8, 29}, {2015, 9, 30, 2015.75, 95, 6.3, 31}}
```

The next rule extracts just the date information and the sunspot number and then wraps the date in `DateList` to put it in a form suitable for the time series functions.

```
In[5]:= test3 = test2 /. {y_Integer, m_, d_, _Real, sn_Integer, __} :> {DateList[{y, m, d}], sn}

Out[5]= {{{1818, 1, 1, 0, 0, 0.}, -1}, {{1818, 1, 2, 0, 0, 0.}, -1},
         {{2015, 9, 29, 0, 0, 0.}, 117}, {{2015, 9, 30, 0, 0, 0.}, 95}}
```

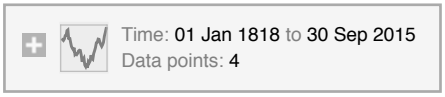
Some pruning of the data will help to better see the patterns in the sunspot activity. From the SILSO website, we find sunspot number values of −1 indicate a missing observation. We will replace those with `Missing[]` which the time series functions will ignore by default.

```
In[6]:= test4 = test3 /. {lis : {__}, sn_ /; sn == -1} :> {lis, Missing[]}

Out[6]= {{{1818, 1, 1, 0, 0, 0.}, Missing[]}, {{1818, 1, 2, 0, 0, 0.}, Missing[]},
         {{2015, 9, 29, 0, 0, 0.}, 117}, {{2015, 9, 30, 0, 0, 0.}, 95}}
```

Finally, the data is converted to a `TimeSeries` object which will simplify visualization and analysis.

```
In[7]:= TimeSeries[test4]

Out[7]= TimeSeries[ 

]
```

Now we repeat the above steps for the entire data set.

```
In[8]:= data2 = data /. lis : {y_, m_, d_, _, _, _, _, _} :> Most[lis];

In[9]:= data3 = data2 /. {y_Integer, m_, d_, _Real, sn_Integer, __} :> {DateList[{y, m, d}], sn};
```

```
In[10]:= data4 = data3 /. {lis : {__}, sn_ /; sn == -1} :> {lis, Missing[]};
```

This is the form we need for time series functions, so we are ready to convert the entire data set into a `TimeSeries` object.

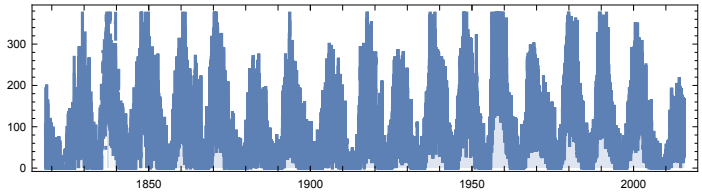
```
In[11]:= tsData = TimeSeries[data4]
```

```
Out[11]= TimeSeries[ Time: 01 Jan 1818 to 30 Sep 2015
                     Data points: 72226 ]
```

The time series is ready to be passed to `DateListPlot` to get a quick visualization of the raw data.

```
In[12]:= DateListPlot[tsData, AspectRatio → 1 / 4, Filling → Bottom]
```

```
Out[12]=
```



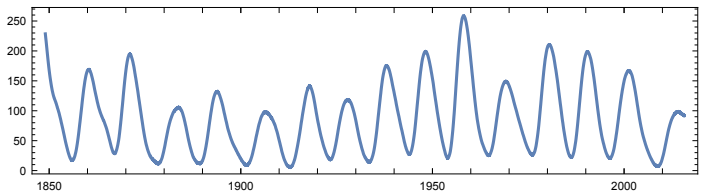
Although the periodic nature of the sunspot activity is already quite visible, some further work is needed: much of the sunspot data is known to be unreliable prior to 1849, so we will start the series at that point (using `TimeSeriesWindow`); and since the observations are relatively uncertain, the data is fairly noisy and so we smooth it using a 24-month Gaussian filter.

```
In[13]:= tsw = TimeSeriesWindow[tsData, {AbsoluteTime[{1849, 1, 1}], AbsoluteTime[]}]
```

```
Out[13]= TimeSeries[ Time: 01 Jan 1849 to 30 Sep 2015
                     Data points: 60903 ]
```

```
In[14]:= gfData = GaussianFilter[tsw, 24 × 30];
         DateListPlot[gfData, AspectRatio → 1 / 4]
```

```
Out[15]=
```



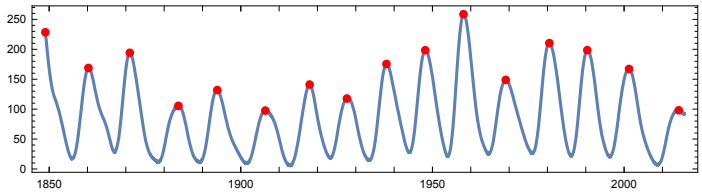
We could go a few steps further and compute the frequency of the peaks. Some experimentation is necessary to find the correct scale for `FindPeaks`.

```
In[16]:= peaks = FindPeaks[gfData, 365]
```

```
Out[16]= TimeSeries[ Time: 01 Jan 1849 to 05 Apr 2014
                     Data points: 16 ]
```

```
In[17]:= DateListPlot[{gfData, peaks}, AspectRatio → 1 / 4, Joined → {True, False},
          PlotStyle → {Automatic, {Red, PointSize[Medium]}}]
```

```
Out[17]=
```



To extract the dates from the time-series peaks and convert to date lists, we need to map `DateList` across these values (`Map` is discussed in the next chapter).

```
In[18]:= dates = Map[DateList, peaks["Times"]]
```

```
Out[18]= {{1849, 1, 1, 0, 0, 0.}, {1860, 3, 29, 0, 0, 0.}, {1871, 1, 18, 0, 0, 0.}, {1883, 9, 19, 0, 0, 0.},
          {1893, 11, 14, 0, 0, 0.}, {1906, 5, 24, 0, 0, 0.}, {1917, 12, 16, 0, 0, 0.}, {1927, 9, 8, 0, 0, 0.},
          {1938, 1, 1, 0, 0, 0.}, {1948, 2, 11, 0, 0, 0.}, {1958, 2, 5, 0, 0, 0.}, {1969, 2, 13, 0, 0, 0.},
          {1980, 7, 2, 0, 0, 0.}, {1990, 5, 24, 0, 0, 0.}, {2001, 4, 16, 0, 0, 0.}, {2014, 4, 5, 0, 0, 0.}}
```

To get the differences between each pair of peaks, we create successive pairs of the dates with `Partition` and then use a transformation rule to pass each pair of dates to `DateDifference`.

```
In[19]:= Partition[dates, 2, 1] /. {d1 : {__}, d2 : {__}} :> DateDifference[d1, d2, "Year"]

Out[19]= {11.2404 yr, 10.8082 yr, 12.6685 yr, 10.1534 yr, 12.5233 yr, 11.5644 yr, 9.72877 yr,
  10.3151 yr, 10.112 yr, 9.98356 yr, 11.0219 yr, 11.3825 yr, 9.89315 yr, 10.8959 yr, 12.9699 yr}
```

And here is the average time span between the cycles, sometimes known as the Schwabe cycle, due to Samuel Heinrich Schwabe who first discovered it in 1843:

```
In[20]:= Mean[%]

Out[20]= 11.0174 yr
```