Outlier detection in a list of numbers

Anton Antonov
Mathematica for Prediction blog
Mathematica for Prediction project at GitHub
October 2013

Introduction

This document is a guide to detecting outliers in one dimensional arrays using the functions of the *Mathematica* package OutlierIdentifiers.m provided by the project MathematicaForPrediction at GitHub, see [1].

We give a brief introduction to outlier detection and then show examples of identifying and plotting outliers.

Outlier detection basics

The purpose of the outlier detection algorithms is to find those elements in a list of numbers that have values significantly higher or lower than the rest of the values.

Taking a certain number of elements with the highest values is not the same as an outlier detection, but it can be used as a replacement.

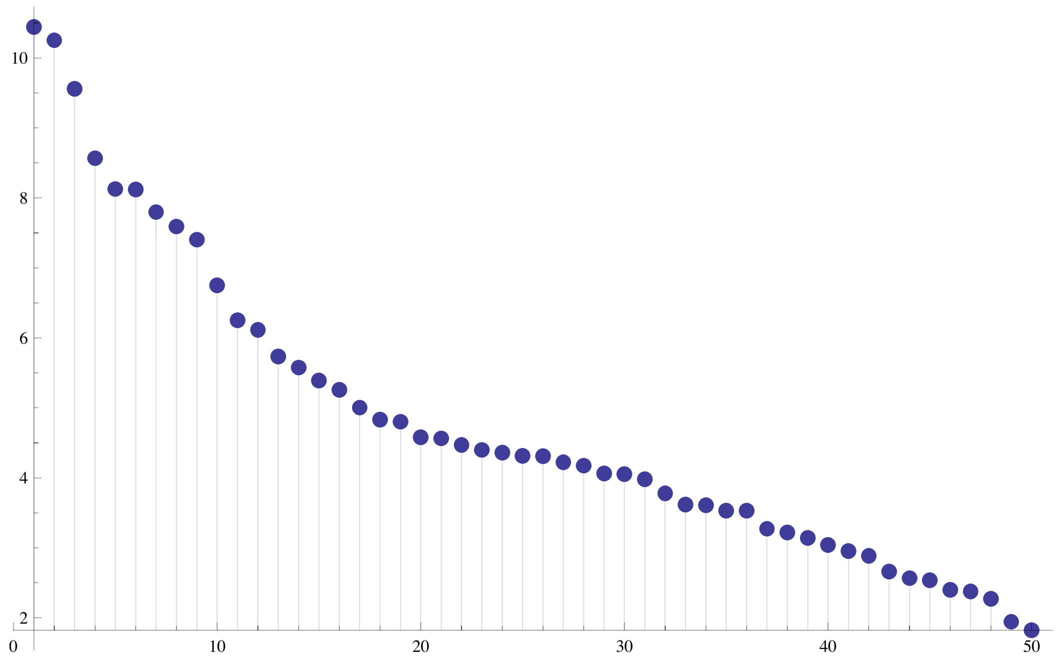
Let us consider the following set of 50 numbers:

```
pnts = RandomVariate[GammaDistribution[5, 1], 50]
```

```
{7.59113, 5.57539, 3.77879, 3.14141, 5.25833, 7.4036, 5.7324, 4.3612,
 3.04052, 4.30872, 7.79725, 8.12759, 6.75185, 4.39845, 8.12002,
 2.95435, 9.55785, 4.06057, 3.27145, 10.2521, 3.52952, 6.25076,
 1.82362, 2.8844, 4.22082, 1.94356, 2.39944, 3.97935, 6.11422,
 4.80236, 8.56683, 3.61667, 2.53776, 3.60869, 2.56662, 4.05317,
 2.27279, 4.46844, 4.57981, 5.00396, 4.17491, 10.4406, 4.31504,
 4.83232, 4.56361, 5.38989, 2.66085, 2.37754, 3.21949, 3.53067}
```

If we sort those numbers descendingly and plot them we get:

```
pnts = pnts // Sort // Reverse;
ListPlot[pnts, PlotStyle → {PointSize[0.015]},
 Filling → Axis, PlotRange → All]
```



Let us use the following outlier detection algorithm:

1. find all values in the list that are larger than the mean value multiplied by 1.5;
2. then find the positions of these values in the list of numbers.

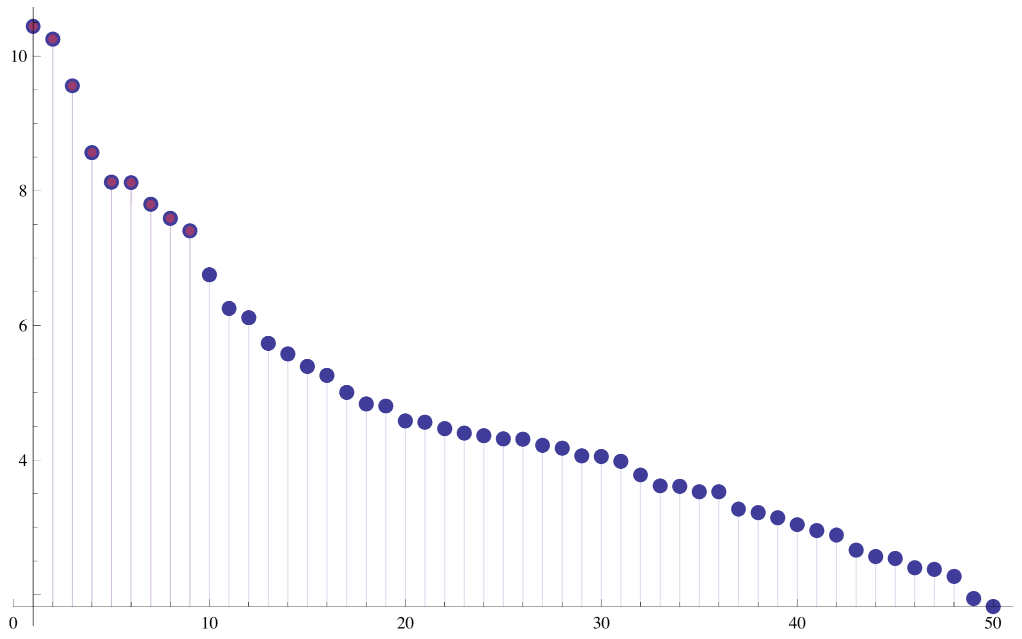
We can implement this algorithm in the following way.

```
pos =
 Flatten[Map[Position[pnts, #] &, Select[pnts, # > 1.5 Mean[pnts] &]]]
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 9}
```

Lets plot all points in blue and the outliers we found in red:

```
ListPlot[{pnts, Transpose[{pos, pnts⟦pos⟧}]},
 PlotStyle → {PointSize[0.015], PointSize[0.009]},
 Filling → Axis, PlotRange → All]
```



Instead of the mean value we can use another reference point, like the median value. Obviously, we can also use a multiplier different than 1.5.

Implementation

First let us load the outlier identification package:

```
Get["~/MathFiles/MathematicaForPrediction/OutlierIdentifiers.m"]
```

We can find the outliers in a list of numbers with the function `OutlierIdentifier`:

```
OutlierIdentifier[pnts, HampelIdentifierParameters]
```

```
{10.4406, 10.2521, 9.55785, 8.56683, 8.12759,
 8.12002, 7.79725, 7.59113, 7.4036, 6.75185, 6.25076,
 2.39944, 2.37754, 2.27279, 1.94356, 1.82362}
```

The package has three functions for the calculation of outlier identifier parameters over a list of numbers

```
HampelIdentifierParameters[pnts]
```

```
{2.50757, 6.11619}
```

```
SPLUSQuartileIdentifierParameters[pnts]
```

```
{-0.549889, 9.50178}
```

```
QuartileIdentifierParameters[pnts]
```

```
{1.79581, 6.82164}
```

Elements of the number list that are outside of the numerical interval made by one of these pairs of numbers are considered outliers.

In many cases we want only the top outliers or only the bottom outliers. We can use the functions `TopOutliers` and `BottomOutliers` for that.

```
OutlierIdentifier[pnts,
 TopOutliers[HampelIdentifierParameters[#]] &]
```

```
{10.4406, 10.2521, 9.55785, 8.56683, 8.12759,
 8.12002, 7.79725, 7.59113, 7.4036, 6.75185, 6.25076}
```

```
OutlierIdentifier[pnts,
 BottomOutliers[HampelIdentifierParameters[#]] &]
```

```
{2.39944, 2.37754, 2.27279, 1.94356, 1.82362}
```

The function `TopOutliers` replaces the lower outlier threshold with $-\infty$. Similarly `BottomOutliers` replaces the upper outlier threshold with ∞.

```
TopOutliers[HampelIdentifierParameters[pnts]]
```

```
{-∞, 6.11619}
```

```
BottomOutliers[HampelIdentifierParameters[pnts]]
```

```
{2.50757, ∞}
```

In many cases we also want the positions of the outliers not the outliers themselves.

```
OutlierPosition[pnts, HampelIdentifierParameters]
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 46, 47, 48, 49, 50}
```

Here is an example of calling all three outlier identifiers in the package [1].

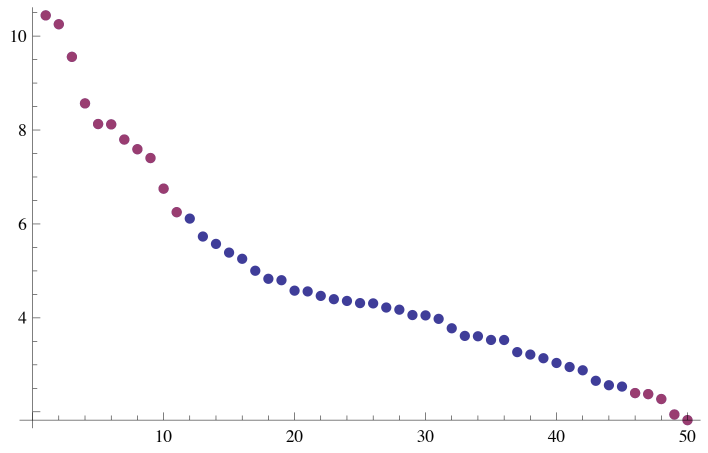
```
Function[{oin}, OutlierIdentifier[pnts, TopOutliers[oin[#]] &]] /@
 {HampelIdentifierParameters,
  SPLUSQuartileIdentifierParameters, QuartileIdentifierParameters}
```

```
{{10.4406, 10.2521, 9.55785, 8.56683, 8.12759,
  8.12002, 7.79725, 7.59113, 7.4036, 6.75185, 6.25076},
 {10.4406, 10.2521, 9.55785}, {10.4406, 10.2521, 9.55785,
  8.56683, 8.12759, 8.12002, 7.79725, 7.59113, 7.4036}}
```

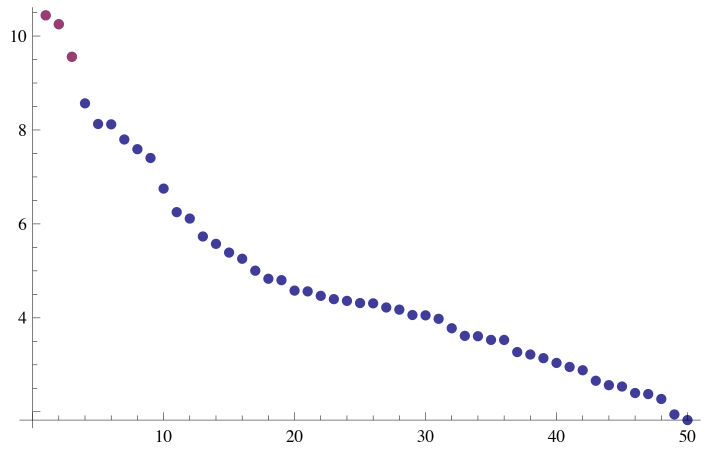
Visualization

The package [1] also provides functions for visualizing the outliers (using `ListPlot`).

```
ListPlotOutliers[pnts, HampelIdentifierParameters,
 PlotStyle → {PointSize[0.015]}]
```



```
ListPlotOutliers[pnts, SPLUSQuartileIdentifierParameters,
 PlotStyle → {PointSize[0.015]}]
```



References

[1] Anton Antonov, Implementation of one dimensional outlier identifying algorithms in *Mathematica*, source code at GitHub, https://github.com/antononcube/MathematicaForPrediction, package OutlierIdentifiers.m, (2013).