binostics: computing scagnostics measures in R and C++

by Ursula Laa, Dianne Cook, Hadley Wickham, and Heike Hofmann

Abstract An abstract of less than 150 words.

Introduction

A scatterplot matrix (SPLOM) is a useful plot to make to examine the distribution of two variables, for linear or nonlinear association, clustering, outliers, discrete, or barriers. However, if there are many variables there will be too many to practically display. For many variables it can be useful to find the most interesting pairs of variables using scagnostics (scatterplot diagnostics) (Wilkinson et al., 2005, Wilkinson and Wills (2008)). A smaller subset of interesting pairs of variables can then be shown in a SPLOM.

There are eight measures calculated in scagnostics: outlying, skewed, sparse, clumpy, striated, convex, skinny, stringy, and monotonic. These, themselves can be examined using a SPLOM. This is especially helpful if some interactivity, so that the scagnostic values can be explored using mouseover or tooltip, to reveal the pair of variables that generates any value.

- scatter plot matrix, need for variable selection with big data
- scagnostics (Wilkinson et al., 2005, Wilkinson and Wills (2008)): characterise scatterplots with 8 summaries (scatterplot diagnostics), select interesting plots from a SPLOM of the scagnostics measures and interactively look at those
- similar methods (e.g. MIC(Reshef et al., 2011), available via minerva package (Albanese et al., 2012))
- alternative methods to see combinations of variables: tours (Asimov, 1985, Wickham et al. (2011)), PCA, LDA, PP (Friedman, 1987)

things to mention: scagnostics package (CRAN) (Urbanek et al., 2012), mbgraphic (Grimm, 2017) (archived!) and Katrins thesis work (Grimm, 2016)

Scagnostics measures

Following Wilkinson et al. (2005) the scagnostics measures are evaluated on the data after hexagon binning, and based on the Delaunay triangulation for comutational efficiency. In addition, outlying points are removed before computing the measures (with exception of the Outlying measure), to make the measures more robust.

Underlying definitions

All measures are based on graphs, i.e. a set of vertices and edges, that can all be extracted from the Delaunay triangulation, namely:

- the convex hull, i.e. the outer edges of the Delaunay triangulation
- the alpha hull (Edelsbrunner et al., 1983)
- the minimum spanning tree (MST) (Kruskal, 1956)

In the following, the length of the MST is defined as the sum of lengths of all edges, and q_x is defined as the xth percentile of the MST edge lengths.

When computing the alpha hull, α is set to q_{90} (Wilkinson et al., 2005). The definition of outlying points is also based on edge length, with points being tagged as outlying if all adjacent edges in the MST have a lenght larger than

$$w = q_{75} + 1.5(q_{75} - q_{25}). \tag{1}$$

Several of the measures are corrected for dependence on sample size using the weight

$$\omega = 0.7 + \frac{0.3}{1+t^2}, \tag{2}$$

where $t = n/500$ and n the sample size. This correction weight was determined by comparing the scagnostics measures over a large number of datasets (Wilkinson et al., 2005).

Measure definitions

- **Outlying**: compares the edge lengths of the MST of outlying points with the length of the original MST T
$$c_{\text{outlying}} = \frac{length(T_{\text{outliers}})}{length(T)} \tag{3}$$
- **Skewed**: is measuring skewness in the distribution of MST edge lengths (and thus large values might not correspond to skewed distributions of points) as
$$c'_{\text{skewed}} = \frac{q_{90} - q_{50}}{q_{90} - q_{10}}, \tag{4}$$
and is corrected to adjust for dependence on the sample size as
$$c_{\text{skewed}} = 1 - \omega(1 - c'_{\text{skewed}}). \tag{5}$$
- **Sparse**: detects if points are only found in small number of locations in the plane, as is the case for categorical variables,
$$c_{\text{sparse}} = \omega q_{90}. \tag{6}$$
- **Clumpy**: to detect clustering we split the MST in two parts by removing a single edge j, and compare the largest edge lenght within the smaller of the two subsets to the length of the removed edge j. The clumpy measure is defined by maximising over all edges in the MST as
$$c_{\text{clumpy}} = \max_j [1 - \max_k [\text{length}(e_k)]/\text{length}(e_j)], \tag{7}$$
with k running over all edges in the smaller subgraph found after removing edge j from the MST.
- **Striated**: aims to detect patterns like smooth algebraic functions or parallel lines, by evaluating the angles between adjacent edges,
$$c_{\text{striated}} = \frac{1}{|V|} \sum_{v \in V^{(2)}} I(\cos\theta_{e(v,a)e(v,b)} < -0.75), \tag{8}$$
where V is the set of vertices, $|V|$ the total number of vertices in the triangulation, $V^{(2)}$ the subset of vertices with two edges (i.e. vertices of degree two), and I the indicator function.
- **Convex**: convexity is computed as the ratio of the area of the alpha hull A and the convex hull H, adjusting for sample size dependence,
$$c_{\text{convex}} = \omega \frac{area(A)}{area(H)}. \tag{9}$$
- **Skinny**: The ratio of the perimeter to the area of the alpha hull A, with normalization such that zero corresponds to a full circle and values close to 1 indicate a skinny polygon,
$$c_{\text{skinny}} = 1 - \frac{\sqrt{4\pi area(A)}}{perimeter(A)}. \tag{10}$$
- **Stringy**: a stringy distribution should have no branches in the MST. This is evaluated by counting the number of vertices of degree two and comparing them to the total number of vertices (dropping those of degree one),
$$c_{\text{stringy}} = \frac{|V^{(2)}|}{|V| - |V^{(1)}|}. \tag{11}$$
- **Monotonic**: monotonicity is evaluated via the squared Spearman correlation coefficient, i.e. the Pearson correlation between the ranks of x and y,
$$c_{\text{monotonic}} = r^2_{\text{Spearman}}. \tag{12}$$

XXX maybe there is a problem with the implementation of outlying, because this sometimes gives values larger than one? (e.g. vs vs am on mtcars, also gives NaN for several other measures: clumpy, striated, stringy, monotonic)

Interface

The elementary function in the binostics package `scagnostics` and can be called with a pair of vectors or a two-dimensional data structure (a matrix or data frame).

The default S3 method is for a pair of vectors and will compute the scagnostics measures for a single scatter plot. In this case additional control and output. The additional arguments `bins` and `outlierRmv` can be used for detailed checkes, but should not be necessary for most applications. The output in this mode is a named list `out`, with `out$s` reporting the scagnostics measures for the scatter plot, and `out$bins` returns the binned data used to compute them. Note that only non-empty bins are kept, and the format is a $3 \times n$ matrix where the columns correspond to the bin position in x, y and the bin count, and each row is a non-empty bin. Internally the x and y axis is mapped to integer numbers, and the bin size is reset if there are more than 250 non-empty bins, and this is reflected in the matrix returned to the user.

In most applications we are interested in the scagnostics measures for all combinations of variables in the input data. This is evaluated when passing a matrix or data frame to the `scagnostics` function. In this mode the function returns a data frame, where each row corresponds to one combination of variables, and we have one column for each scagnostics index. Two additional columns `var1` and `var2` report the corresponding variable names.

Finally, we may only be interested in a single scagnostics index for a pair of vectors. Since the measures all rely on the underlying binning and triangulation we simply provide wrapper functions to access this information conveniently. These functions are named according after the corresponding scagnostics measure and report the index as an unnamed scalar. This mode would be preferred e.g. when using the measure as a projection pursuit index.

Implementation connecting R and C++

The interface is written in R and handels the reading the input, finds all combinations of variables in the case of matrix input. For each pair of variables it then pre-processes the data: entries with missing values in either of the two variables are dropped, and each variable is centered and scaled by its range. It is then using a direct call to C++ for the computation of the measures, as described below, and finally collecting and formatting the output as described above.

The underlying C++ implementation allows for a fast evaluation, in particular it handles the binning and triangulation. The triangulation is used to determine the MST, convex hull and alpha hull, which are then used to compute the measures. The calculation is done in the following steps:

- Binning of the data: we use hexagonal binning, where the number of bins is a free parameter. Note that if this results in more than 250 non-empty bins in the result, half the number of bins will be used instead.
- Computing the Delaunay triangulation and the MST
- Outlier detection: we use a cutoff on the edge length (see Eq. 1), to identify outlying points. The triangulation and computation of the MST are repeated after removing the tagged outliers.
- Compute scagnostics measures.

Example

To illustrate the use of scagnostics via the binostics package, we evaluate them for the Parkinsons data (Little et al., 2007) available from Little. The dataset is composed of 22 biomedical voice measurements from 31 people, 23 with Parkinson's disease (PD), each column is a particular voice measure, and each row corresponds one of 195 voice recording from these individuals. The original aim of the data was to discriminate healthy people from those with PD, and the data also records to "status" (0 for healthy individuals and 1 for those with PD). Here we instead look for patterns in combinations of the 22 variables, irrespective of the status.

Given the 22 input variables, there are 231 bivariate plots to look at, and we can use scagnostics to select the most interesting ones. We start by reading in the data, dropping the identifier and status columns, and evaluating the scagnostics measures for all variable combinations. To show the data structure we use the `glimpse` function to show the first few entries. The data frame returned by the `binostics::scagnostics` call has one column for each scagnostics measure, and `var1` and `var2` reporting the corresponding variable combination for each entry.

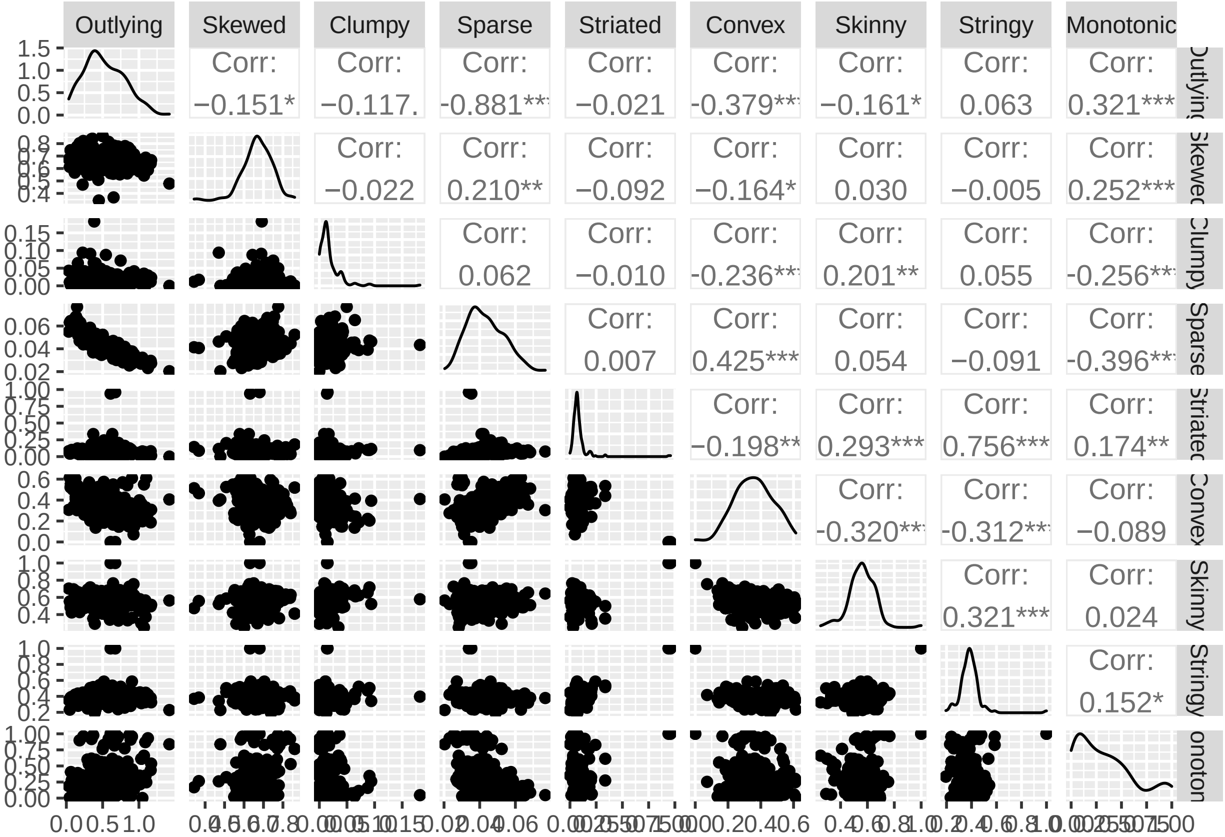
```
pk <- as.data.frame(read.csv("data/parkinsons.csv")[,-c(1,18)])
s <- binostics::scagnostics(pk)
tibble::glimpse(s)
```

```
#> Rows: 231
#> Columns: 11
#> $ Outlying  <dbl> 0.9257279, 0.3293567, 0.4035949, 0.4507585, 0.8905294, 0....
#> $ Skewed    <dbl> 0.6277309, 0.6886678, 0.7286064, 0.6905160, 0.6125908, 0....
#> $ Clumpy    <dbl> 0.0110226189, 0.0907209155, 0.0637457396, 0.0011010779, 0...
#> $ Sparse    <dbl> 0.03128291, 0.04721585, 0.04550484, 0.04338813, 0.0301569...
#> $ Striated  <dbl> 0.04716981, 0.09848485, 0.06896552, 0.07857143, 0.1071428...
#> $ Convex    <dbl> 0.07225233, 0.20414030, 0.13634988, 0.31400790, 0.1443526...
#> $ Skinny    <dbl> 0.7542742, 0.7168408, 0.6808071, 0.6612432, 0.6114717, 0....
#> $ Stringy   <dbl> 0.4168126, 0.5049218, 0.3669219, 0.4064421, 0.3517199, 0....
#> $ Monotonic <dbl> 0.25289570, 0.34529197, 0.02725372, 0.06387947, 0.0296434...
#> $ var1      <chr> "MDVP.Fo.Hz.", "MDVP.Fo.Hz.", "MDVP.Fhi.Hz.", "MDVP.Fo.Hz...
#> $ var2      <chr> "MDVP.Fhi.Hz.", "MDVP.Flo.Hz.", "MDVP.Flo.Hz.", "MDVP.Jit...
```

To get an overview of the results we can make a scatterplot matrix showing pairs of scagnostics measures, where each point now corresponds to one combination of input variables (i.e. one bivariate scatterplot of the original data). This is most useful in an interactive plot that allows us to identify the variables associated with each point, and we can combine the `GGally::ggpairs` with `plotly::ggplotly` to generate this interactive display, as shown in the commented section of the code.

We can generate a scatterplot matrix directly using the data frame returned by `binostics`, where the first nine columns contain the scagnostics measures.

```
# get SPLOM using GGally
GGally::ggpairs(s, columns=1:9)
```



This shows a few outlying points, i.e. parameter combinations that should correspond to interesting scatterplots of the data. To find the corresponding parameters we can use `plotly` in an interactive display. To display the labels when hovering over points we can set the text aestetics, note however that this will be interpreted as grouping and will result in warnings when generating the graph.

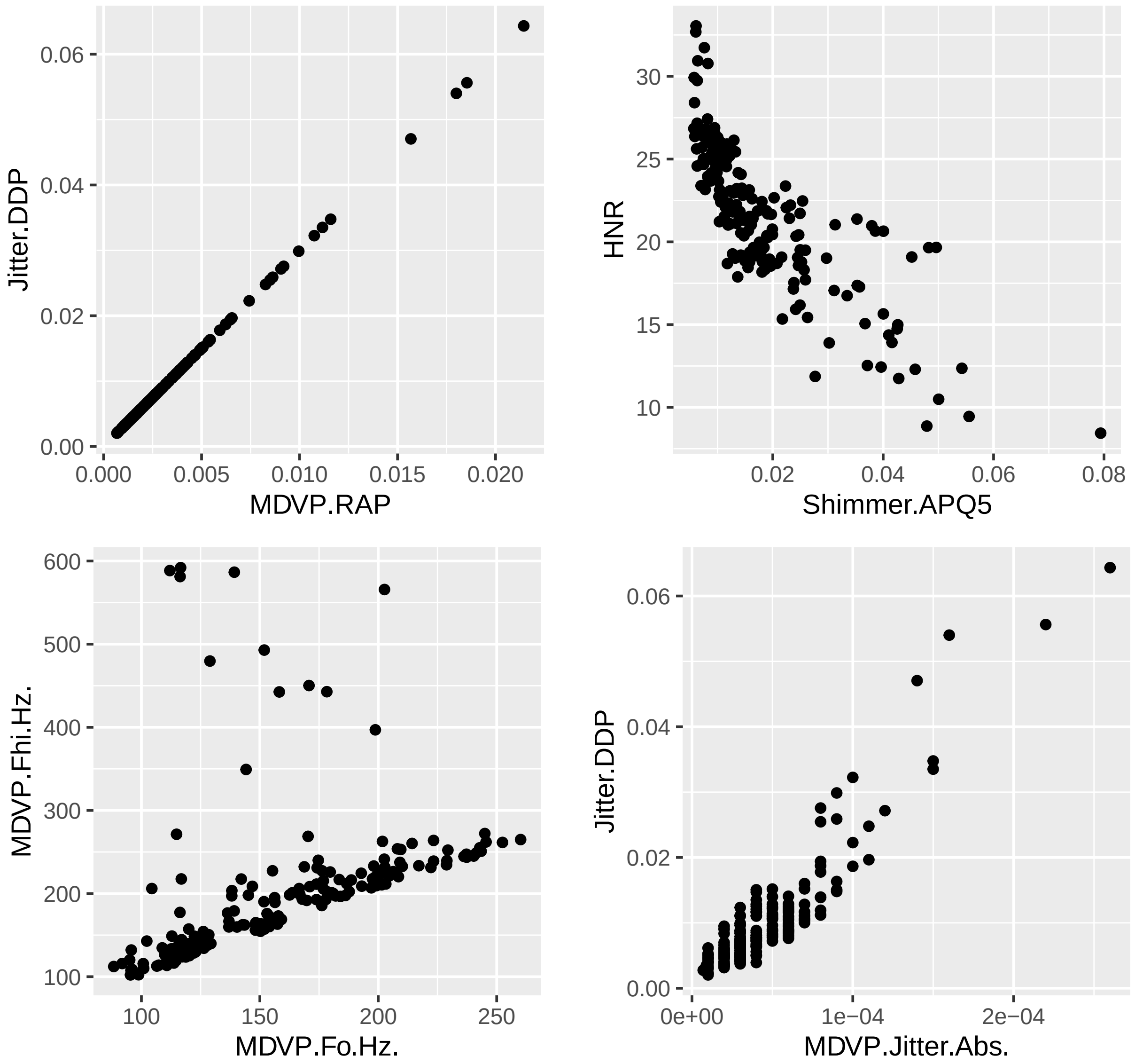
XXX additionally would be useful to have linked highlighting when hovering so we can compare across scagnostics measures

```
## read variable columns to generate point labels
## these are only used in the interactive display
```

```
s <-  dplyr::mutate(s, labs = paste0(var1,"-",var2))
## to generate a useful interactive plotly version we can
## pass in the labels as text text (note that this will cause warnings in ggpairs)
GGally::ggpairs(s, aes(text = labs), columns=1:9) %>%
  plotly::ggplotly()
```

Some interesting combinations that we have identified are:

- “MDVP.RAP” vs “Jitter.DDP”: this combination leads to values close to or exactly one for multiple measures, Striated, Skinny, Stringy and Monotonic, while Convex, Clumpy and Sparse are close to zero. However, the scatterplot is not very interesting, the two variables are perfectly correlated.
- “Shimmer.APQ5” vs “HNR”: this combination stand out because it has large Skewed measure (0.84) but lower Monotonic measure (0.53). The scatterplot display shows a branching of observations for larger values of Shimmer.APQ5.
- “MDVP.Fo.Hz.” vs “MDVP.Fhi.Hz.”: this combination has both low Convex (0.07) and Monotonic (0.25) measures, and the scatterplot shows strong linear correlation with few outlying points.
- “MDVP.Jitter.Abs.” vs “Jitter.DDP”: this combination stands out as having an unusual combination of measures, including values of Skewed (0.86), Sparse (0.05), Striated (0.18), Skinny (0.41) and Monotonic (0.76). The scatterplot display shows discreetness along MDVP.Jitter.Abs. and an unusual shape of the distribution.



Summary

- scagnostics measures useful when exploring large datasets
- the binostics implementaton is efficient thanks to c++ interface, and portable (no java dependence)
- most useful for interactive exploration (maybe good to use in Shiny app?)
- connection with PP (cite PPI paper)

Bibliography

D. Albanese, M. Filosi, R. Visintainer, S. Riccadonna, G. Jurman, and C. Furlanello. minerva and minepy: a C engine for the MINE suite and its R, Python and MATLAB wrappers. *Bioinformatics*, 29 (3):407–408, 12 2012. ISSN 1367-4803. [p1]

D. Asimov. The Grand Tour: A Tool for Viewing Multidimensional Data. *SIAM Journal of Scientific and Statistical Computing*, 6(1):128–143, 1985. [p1]

H. Edelsbrunner, D. Kirkpatrick, and R. Seidel. On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*, 29(4):551–559, 1983. [p1]

J. H. Friedman. Exploratory projection pursuit. *Journal of the American Statistical Association*, 82:249–266, 1987. [p1]

K. Grimm. *Kennzahlenbasierte Grafikauswahl*. doctoral thesis, Universität Augsburg, 2016. [p1]

K. Grimm. *mbgraphic: Measure Based Graphic Selection*, 2017. URL https://CRAN.R-project.org/package=mbgraphic. R package version 1.0.0. [p1]

J. B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society*, 7(1):48–50, 1956. [p1]

M. A. Little. Oxford parkinson's disease detection dataset. https://archive.ics.uci.edu/ml/datasets/Parkinsons. [p3]

M. A. Little, P. E. McSharry, S. J. Roberts, D. A. Costello, and I. M. Moroz. Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection. *BioMedical Engineering OnLine*, 6(1):23, 2007. doi: 10.1186/1475-925X-6-23. URL https://doi.org/10.1186/1475-925X-6-23. [p3]

D. N. Reshef, Y. A. Reshef, H. K. Finucane, S. R. Grossman, G. McVean, P. J. Turnbaugh, E. S. Lander, M. Mitzenmacher, and P. C. Sabeti. Detecting novel associations in large data sets. *Science*, 334(6062): 1518–1524, 2011. ISSN 0036-8075. [p1]

S. Urbanek, L. Wilkinson, and A. Anand. scagnostics: Compute scagnostics - scatterplot diagnostics. https://cran.r-project.org/web/packages/scagnostics/index.html, 2012. [p1]

H. Wickham, D. Cook, H. Hofmann, and A. Buja. tourr: An R package for exploring multivariate data with projections. *Journal of Statistical Software*, 40(2):1–18, 2011. [p1]

L. Wilkinson and G. Wills. Scagnostics distributions. *Journal of Computational and Graphical Statistics*, 17 (2):473–491, 2008. [p1]

L. Wilkinson, A. Anand, and R. Grossman. Graph-theoretic scagnostics. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, pages 157–164, Oct 2005. [p1]

Ursula Laa
Affiliation
line 1
line 2

ursula.laa@boku.ac.at

Dianne Cook
Monash University
Department of Econometrics and Business Statistics

dicook@monash.edu

Hadley Wickham
RStudio, PBC

hadley@rstudio.com

Heike Hofmann
Iowa State University
Department of Statistics

hofmann@iastate.edu