A Visualization Tool for Mining Large Correlation Tables: The Association Navigator

Andreas Buja, Abba Krieger, Ed George
Statistics Department, The Wharton School, University of Pennsylvania*

August 22, 2016

1 Overview

The **Association Navigator** is an interactive visualization tool for viewing large tables of correlations. The basic operation is zooming and panning of a table that is presented in graphical form, here called a "blockplot".

The tool is really a tool box that includes, among other things: (1) display of p-values and missing value patterns in addition to correlations, (2) mark-up facilities to highlight variables and sub-tables as landmarks when navigating the larger table, (3) histograms/barcharts, scatterplots and scatterplot matrices as "lenses" into the distributions of variables and variable pairs, (4) thresholding of correlations and p-values to show only strong and highly significant p-values, (5) trimming of extreme values of the variables for robustness, (6) "reference variables" that stay in sight at all times, and (7) wholesale adjustment of groups of variables for other variables.

The tool has been applied to data with nearly 2,000 variables and associated tables approaching a size of 2,000×2,000. The usefulness of the tool is less in beholding gigantic tables in their entirety and more in searching for interesting association patterns by navigating manageable but numerous and interconnected sub-tables.

2 Introduction

This document describes the **Association Navigator** (***AN*** for short) in three sections: (1) In this introductory Section 2 we give some background about the data analytic and

*This work was partially supported by a grant from the Simons Foundation (SFARI awards #121221 and #296012 to A.K.). We appreciate obtaining access to the phenotypic data on SFARI Base (https://base.sfari.org). Partial support was also provided by NSF Grant DMS-1007689 to A. Buja.

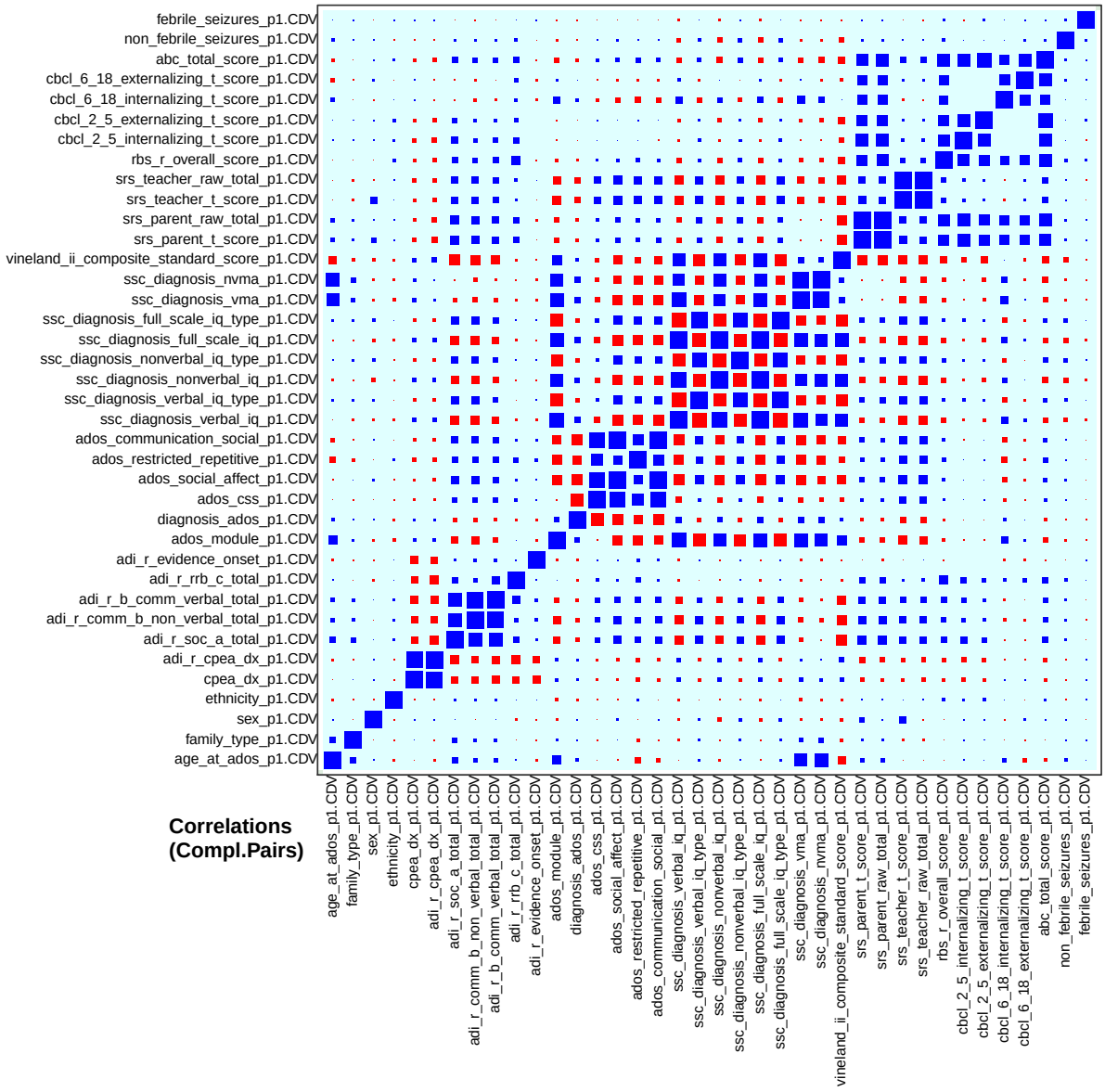

Figure 1: *A first example of a "blockplot": labels in the bottom and left margins show variable names, and blue and red blocks in the plotting area show positive and negative correlations.*

statistical problem addressed by this tool; (2) in Section 3 we describe the graphical displays used by the tool; (3) in Section 4 we describe the actual operation of the tool. We start with some background:

An important focus of contemporary statistical research is on methods for large multivariate data. The term "large" can have two meanings, not mutually exclusive: (1) a large number of cases (records, rows), also called the "large-n problem", or (2) a large number of variables (attributes, columns), also called the "large-p problem." The two types of largeness call for different data analytic approaches and determine the kinds of questions that can be answered by the data. Most fundamentally it should be observed that increasing n, the number of cases, and increasing p, the number of variables, each has very different and in some ways opposite effects on statistical analysis. Since the general multivariate analysis problem is to make statistical inference about the association among variables, increasing n has the effect of improving the certainty of inference due to improved precision of estimates, whereas increasing p has the contrary effect of reducing the certainty of inference due to the multiplicity problem or, more colorfully, the "data dredging fallacy." Therefore the level of detail that can be inferred about association among variables improves with increasing n but it plummets with increasing p.

The problem we address here is primarily the large-p problem. From the above discussion it follows that, for large p, associations among variables can generally be inferred only to a low level of detail and certainty. Hence it is sufficient to measure association by simple means such as plain correlations. Correlations indicate the basic directionality in pairwise association, and as such they answer the simplest but also most fundamental question: are higher values in X associated with higher or lower values in Y, at least in tendency?

Reliance on correlations may be subject to objections because they seem limited in their range of applicability for several reasons: (1) they are considered to be measures of linear association only, (2) they describe bivariate association only, and (3) they apply to quantitative variables only. In Appendix A we refute or temper each of these objections by showing (1) that correlations are usually useful measures of directionality even when the associations are non-linear, (2) that higher-order associations play a reduced role especially in large-p problems, and (3) that with the help of a few tricks of the trade ("scoring" and "dummy coding") correlations are useful even for categorical variables, both ordinal and nominal. In view of these arguments we proceed from the assumption that correlation tables, when used creatively, form quite general and powerful summaries of association among many variables.

In the following sections we describe first how we graphically present large correlation tables and second how we navigate and search them interactively. The software written to this end, the **Association Navigator** or ***AN***, implements the essential displays and interactive functionality to support the "mining" of large correlation tables. The ***AN*** software is written entirely in the **R** language[1].

All data examples in this document are drawn from the phenotypic data in the "Simons Simplex Collection" (SSC) created by the *Simons Foundation Autism Research Initiative* (SFARI). Approved researchers can obtain the SSC dataset used in this document by apply-

[1] http://www.cran.r-project.org

ing at https://base.sfari.org.

3 Graphical Displays

3.1 Graphical Display of Correlation Tables: Blockplots

Figure 1 shows a first example of what we call a "blockplot"[2] of a dataset with $p = 38$ variables. This plot is intended as a direct and fairly obvious translation of a numeric correlation table into visual form. The elements of the plot are as follows:

- The **labels** in the bottom and left margins show line-ups of the same 38 variables: `age_at_ados_p1.CDV`, `family_type_p1.CDV`, `sex_p1.CDV`,.... In contrast to tables, where the vertical axis lists variables top down, we follow the convention of scatterplots where the vertical axis is ascending and hence the variables are listed bottom up.

- The blue and red squares or "**blocks**" represent the pairwise correlations between variables at the intersections of the (imagined) horizontal and vertical lines drawn from the respective margin labels. The magnitude of a correlation is reflected in the **size** of the block and its sign in the **color**: positive correlations are shown in blue and negative correlations in red.[3] — Along the ascending 45-degree **diagonal** are the correlations +1 of the variables with themselves, hence these blocks are of maximal size. The closeness of other correlations to +1 or −1 can be gauged by a size comparison with the diagonal blocks.

- Finally, the plot shows a small comment in the bottom left, "Correlations (Compl.Pairs)", indicating that what is represented by the blocks is correlation of complete — that is, non-missing — pairs of values of the two variables in question. This comment refers to the missing values problem and to the fact that correlation can only be calculated from the cases where the values of both variables are non-missing. The comment also alludes to the possibility that very different types of information could be represented by the blocks, and this is indeed made use of by the ***AN*** software (see Sections 3.3 and 3.4).

As a "reading exercise" consider Figure 2: This is the same blockplot as in Figure 1, but for ease of pointing we marked up two variables on the horizontal axis[4]:

`age_at_ados_p1.CDV`, `ados_restricted_repetitive_p1.CDV`,

[2] This type of plot is also called "fluctuation diagram" (Hofmann, 2000). The term "blockplot" is ours, and we introduce it because it is more descriptive of the plot's visual appearance. We may even dare propose that "blockplot" be contracted to "blot", which would be in the tradition of contracting "scatterplot matrix" to "splom" and "graphics object" to "grob".

[3] We follow the convention from finance where "being in the red" implies negative numbers; the opposite convention is from physics where red symbolizes higher temperatures. Users can easily change the defaults for blockplots; see the programming hints in Appendix B.

[4] This dataset represents a version of the table `proband_cdv.csv` in Version 9 of the phenotypic SSC. The acronym `cdv` means "core descriptive variables".

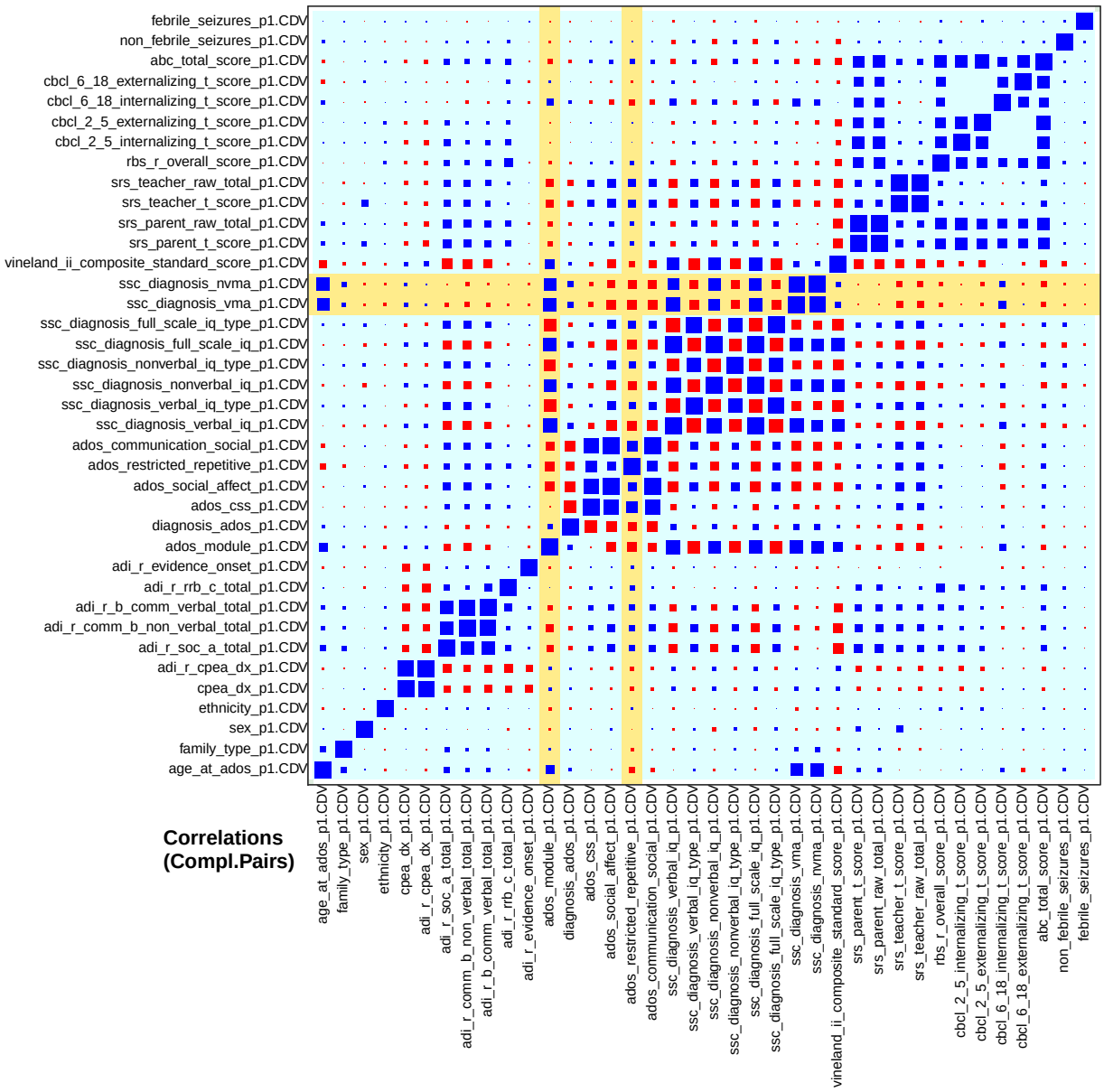

Figure 2: *A "reading exercise" illustrated with the same example as in Figure 1. The salmon-colored strips highlight the variables* `age_at_ados_p1.CDV` *and* `ados_restricted_repetitive_p1.CDV` *on the horizontal axis, and the variables* `ssc_diagnosis_vma_p1.DCV` *and* `ssc_diagnosis_nvma_p1.DCV` *on the vertical axis. At the intersections of the strips are the blocks that reflect the respective correlations.*

meaning "age at the time of the administration of the ADOS or Autism Diagnostic Observation Schedule", and "problems due to restricted and repetitive behaviors", respectively. Two other variables are marked up on the vertical axis:

`ssc_diagnosis_vma_p1.DCV`, `ssc_diagnosis_nvma_p1.DCV`.

meaning "verbal mental age", and "non-verbal mental age", respectively, which are related to notions of IQ. For readability we will shorten the labels in what follows.

As for the actual reading exercise, in the intersection of the left vertical strip with the horizontal strip, we find two blue blocks of which the lower is recognizably larger than the upper (the reader may have to zoom in if viewing the figure in a PDF reader), implying that the correlation of `age_at_ados..` with both `..vma..` and `..nvma..` is positive, but more strongly with the former than the latter, which may be news to the non-specialist: verbal skills are more strongly age-related than non-verbal skills. (Strictly speaking we can claim this only for the present sample of autistic probands.) — Similarly, following the right vertical strip to the intersection with the horizontal strip, we find two red blocks of which again the lower block is slightly larger than the upper, but both are smaller than the blue blocks in the left strip. This implies that `..restricted_repetitive..` is negatively correlated with both `..vma..` and `..nvma..`, but more strongly with the former, and both are more weakly correlated with `..restricted_repetitive..` than with `age_at_ados...` All of this makes sense in light of the apparent meanings of the variables: Any notion of "mental age" is probably quite strongly and positively associated with chronological age; with hindsight we may also accept that problems with specific behaviors tend to diminish with age, but the association is probably less strong than that between different notions of age.

Some other patterns are quickly parsed and understood: The two 2×2 blocks on the upper right diagonal stem from two versions of the same underlying measurements: "raw_total" and "t_score". Next, the alternating patterns of red and blue in the center indicate that the three IQ measures ("verbal", "nonverbal", "full_scale") are in an inverse association with the corresponding IQ types. This makes sense because IQ types are dummy variables that indicate whether an IQ test suitable for cognitively highly impaired probands was applied. — It becomes apparent that one can spend a fair amount of time wrapping one's mind around the visible blockplot patterns and their meanings.

3.2 Graphical Overview of Large Correlation Tables

Figure 1 shows a manageably small set of 38 variables which is yet large enough that the presentation of a numeric table would be painful for anybody. This table, however, is a small subset of a larger dataset of 757 variables which is shown in Figure 3. In spite of the very different appearance, this, too, is a blockplot drawn by the same tool and by the same principles, with some allowance for the fact that $757^2 = 573,049$ blocks cannot be sensibly displayed on screens with an image resolution comparable to 757^2. When blocks are represented by single pixels, blocksize variation is no longer possible. In this case, the tool displays only a selection of correlations that are largest in magnitude. The default (which can be changed) is to show 10,000 of the most extreme correlations, and it is these that give the blockplot in Figure 3 the characteristic pattern of streaks and rectangular concentrations.

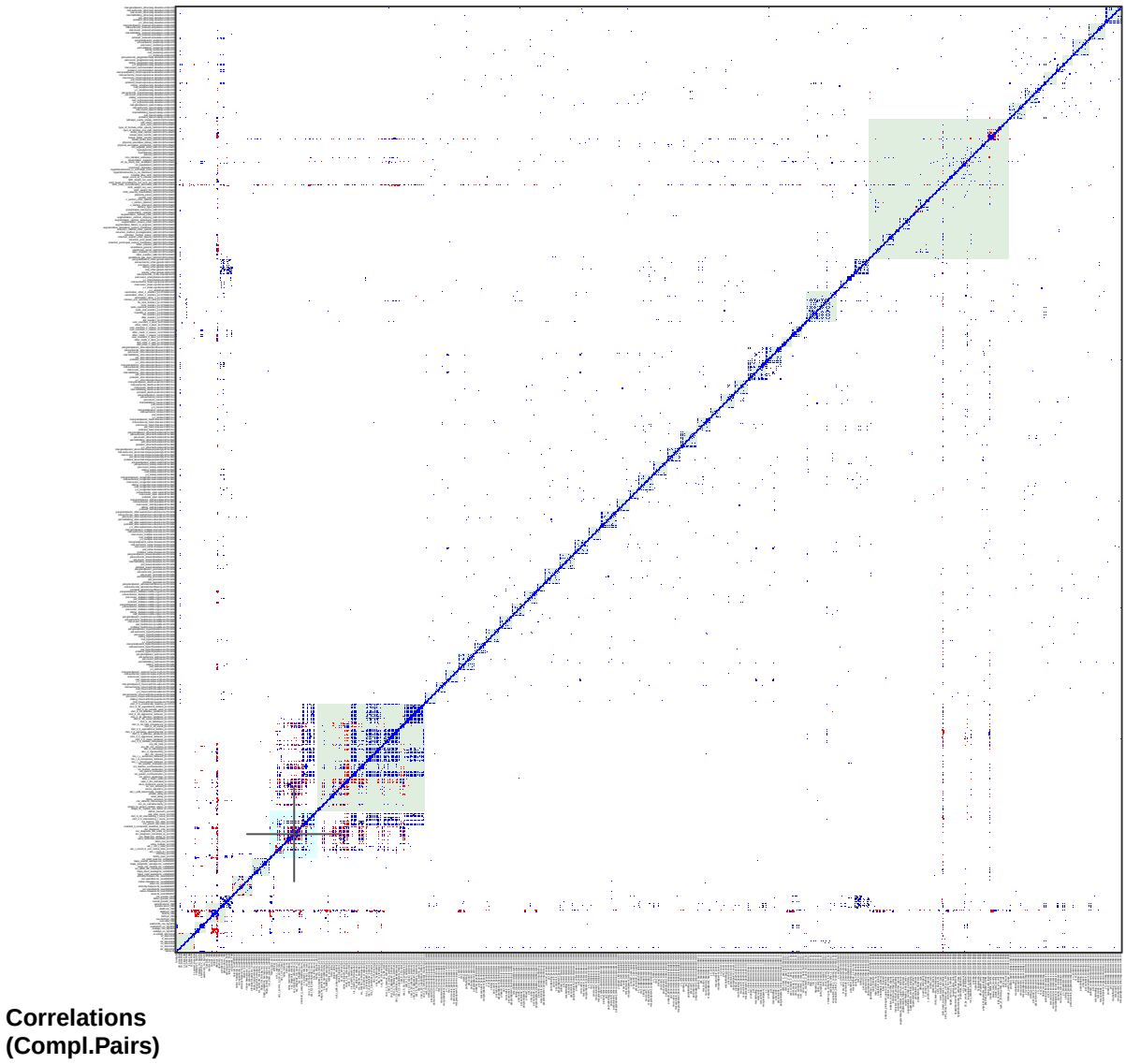

Figure 3: *An overview blockplot of 757 variables. Groups of variables are marked by background highlight squares along the ascending diagonal. The blockplot of Figure 1 is contained in this larger plot and can be found in the small highlight square in the lower left marked by a faint crosshair. Readers who are viewing this document in a PDF reader may zoom in to verify that this highlight square contains an approximation to Figure 1.*

The function of this blockplot is not so much to facilitate discovery as to provide an overview organized in such a way that meaningful subsets are recognizable to the expert who is knowledgeable about the dataset. The tool helps in this regard by providing a way to group the variables and showing the variable groups as diagonal highlight squares. In Figure 3 two large highlight squares are recognizable near the lower left and the upper right. Closer scrutiny should allow the reader to recognize many more much smaller highlight squares up and down the diagonal, each marking a small variable group. In particular, the reader should be able to locate the third-largest highlight square in the lower left, shown in turquoise as opposed to gray and pointed at by a faint crosshair: this square marks the group of 38 variables shown in Figures 1 and 2.

The mechanism by which variable grouping is conveyed to the ***AN*** is a naming convention for variable names: to define a variable group, the variables to be included must be given names that end in the same suffix separated by an underscore "_" (default, can be changed), and the variables must be contiguous in the order of the dataset. As an example, Figure 4 shows the 38 variable group of Figure 1 in the context of its neighbor groups: This group is characterized by the suffix "`p1.CDV`", whereas the neighbor group on the upper right (only a small part is visible) has the suffix "`p1.OCUV`" and the two groups on the lower left have suffixes "`cuPARENT`" and "`racePARENT`".[5] The background highlight squares cover the intra-group correlations for the respective variable groups. As in Figure 3, the highlight square for the 38 variable group is shown in turquoise whereas the neighboring highlight squares are in gray.

Figures 1-4 are a prelude for the zooming and panning functionality to be described in Section 4.

3.3 Other Uses of Blockplots (1): P-Values

Associated with correlations are other quantities of interest that can also be displayed with blockplots, foremost among them the p-values of the correlations. A p-value in this case is a measure of evidence *in favor of* the assumption that the observed correlation is spurious, that is, its deviation from zero is due to chance alone while the population correlation is zero.[6] P-values are hypothetical probabilities, hence they fall in the interval $[0, 1]$. As p-values represent evidence *in favor of* the assumption that *no* linear association exists, it is small p-values that are of interest, because they indicate that the chance of a spurious detection of linear association is small. By convention one is looking for p-values at least below 0.05, for a "Type I error" probability of one in twenty or less. When considering p-values of many correlations on the same dataset — as is the case here — one needs to protect against "multiplicity", that is, the fact that 5% of p-values will be below 0.05 even if in truth all

[5] These suffixes abbreviate the following full-length meanings: "proband 1, core descriptive variables", "proband 1, other commonly used variables", "commonly used for parents" and"race of parents". These variable groups are from the following SSC tables: `proband_cdv.csv`, `proband_ocuv.csv`, and `parent.csv`.

[6] Technically, the (two-sided) p-value of a correlation is the hypothetical probability of observing a future sample correlation greater in magnitude than the sample correlation observed in the actual data — assuming that in truth the population correlation is zero.

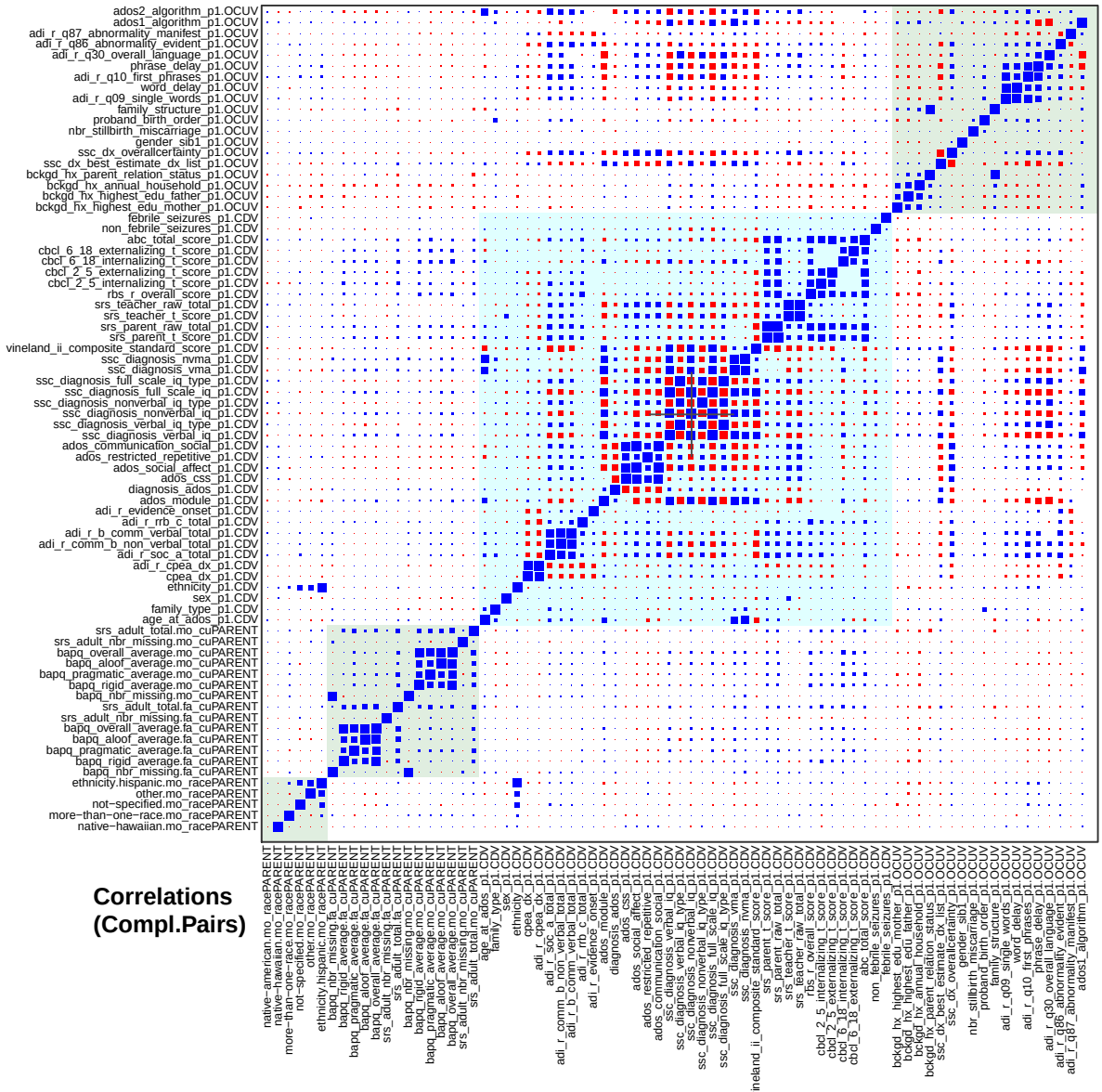

Figure 4: *The 38 variable group of Figure 1 in the context of the neigboring variable groups.*

population correlations vanish. Such protection is provided by choosing a threshold much smaller than 0.05, by the conservative Bonferroni rule as small as 0.05/#correlations. In the data example with 757 variables, the number of correlations is 286,146, hence one might want to choose the threshold on the p-values as low as .05/286,146 or about 1.75 in 10 million. The point is that in large-p problems one is interested in *very small* p-values.[7]

P-values lend themselves easily to graphical display with blockplots, but the direct mapping of p-value to blocksize has some drawbacks. These drawbacks, on the other hand, can be easily fixed:

- P-values are blind to the sign of the correlation: correlation values of +0.95 and -0.95, for example, result in the same two-sided p-value. We correct for this drawback by showing p-values of negative correlations in red color.

- Of interest are small p-values that correspond to correlations of large magnitude, hence a direct mapping would represent the interesting p-values by small blocks, which is visually incorrect because the eye is drawn to large objects, not to large holes. We therefore invert the mapping and associate blocksize with the complement 1−(p-value).

- Drawing on the preceding discussion, our interest is really in very small p-values, and one may hence want to ignore p-values greater than 0.05 altogether in the display. We therefore map the interval [0, 0.05] inversely to blocksize, meaning that p-values below but near 0.05 are shown as small blocks and p-values very near 0.00 as large blocks.

The resulting p-value blockplots are illustrated in Figure 5. The two plots show the same 38 variables group as in Figure 1 with p-values truncated at 0.05 and at 0.000,000,1, respectively, as shown near the bottom left corners of the plots. The p-values are calculated using the usual normal approximation to the null distribution of the correlations. In view of the large sample size, $n \geq 1,800$, the normal approximation can be assumed to be quite good, even though one is going out on a limb when relying on normal tail probabilities as small as 10^{-7}. Then again, p-values this small are strong evidence against the assumption that the correlations are spurious.

3.4 Other Uses of Blockplots (2): Fraction of Missing and Complete Pairs of Values

Missing values are so common that they require special attention and special tools for understanding their patterns. Missing values are sometimes approached with imputation methods, but in view of the large number of variables we wish to explore we use simple deletion methods that rely on the largest number of available values. For correlations this means that we use for a given pair of variables the full set of complete pairs of values. Another common and more stringent deletion method is to use only cases that are complete on all variables,

[7] The letters 'p' in "large-p" and "p-value" bear no relation. In the former, p is derived from "parameter", in the latter from "probability".

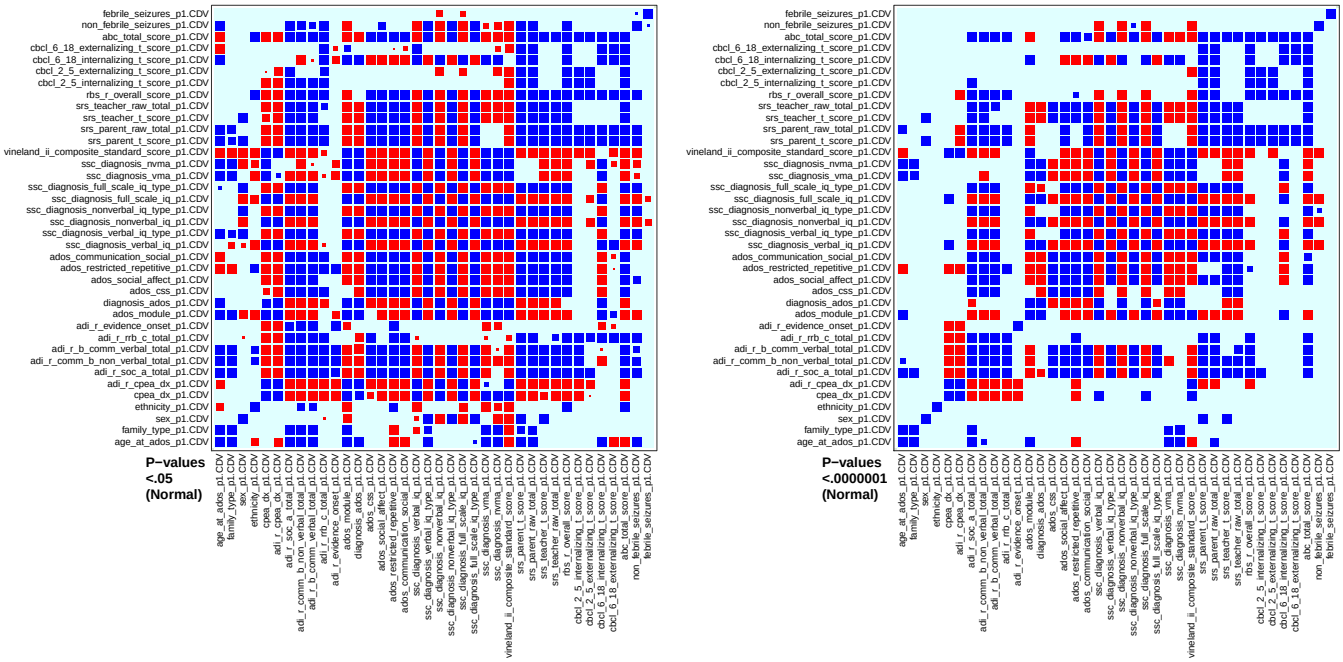

Figure 5: *Blockplots of the p-values for the 38 variable group of Figure 1. Smaller and hence statistically more significant p-values are shown as larger blocks. The colors are inherited from the correlations to reflect their signs.*
Truncation levels of p-values: Left ≥ 0.05*; right* $\geq 0.000,000,1$.
Many modest correlations are extremely statistically significant due to $n \geq 1,800$.

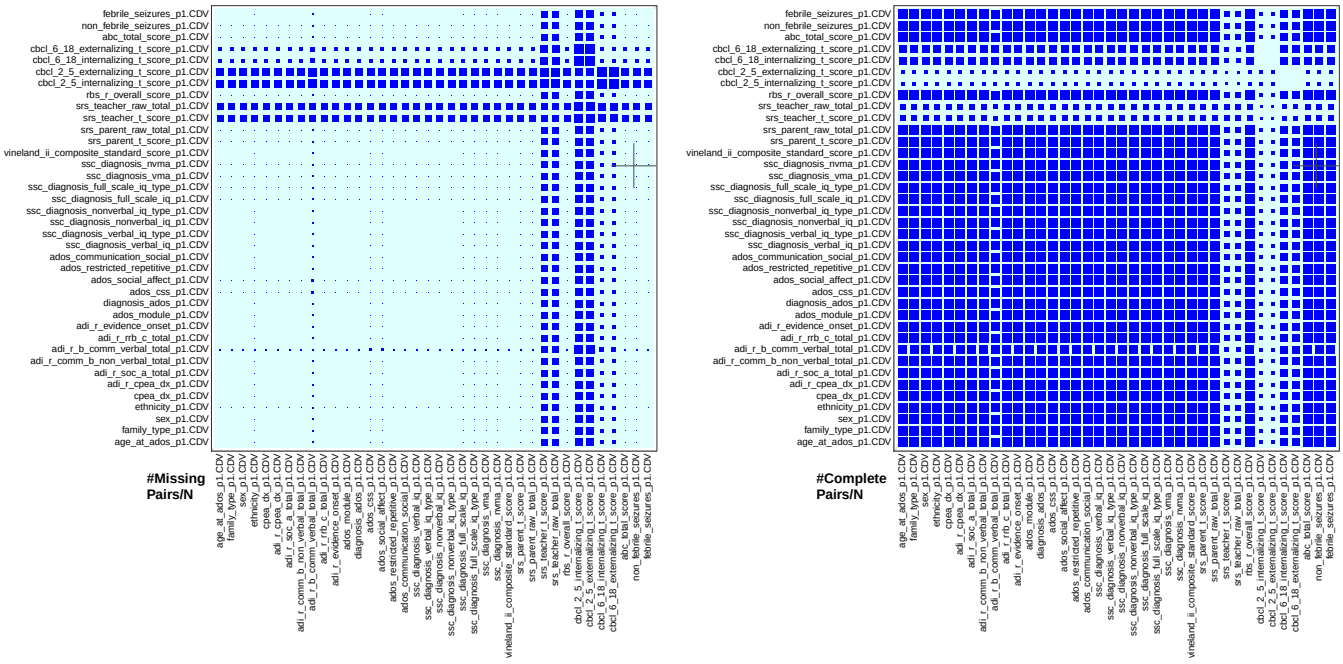

Figure 6: *Blockplots of the fractions of missing (left) and complete (right) pairs of values.*

but in the large-p problem this is not a viable approach because complete cases may well not exist when the number of variables reaches into the hundreds or thousands.

An issue with calculating correlations from maximal sets of complete pairs of values is that this set may vary from correlation to correlation because it is formed from the overlap of non-missing values in both variables. Thus, associated with each correlation $r(x, y)$ are

- the number $n(x, y)$ $(\leq n)$ of complete pairs from which $r(x, y)$ is calculated, and
- the number $m(x, y) = n - n(x, y)$ of incomplete pairs where at least one of the two, x or y, is missing.

Just like the correlations $r(x, y)$, the values $n(x, y)$ and $m(x, y)$ form $n \times n$ tables, hence can be easily visualized with blockplots in their fractional forms $n(x, y)/n$ and $m(x, y)/n$. An example of each is shown in Figure 6, again for the same 38 variable group of Figure 1. Apparently four variables have a major missing value problem.

Depending on whether the number of complete or incomplete pairs dominates, one or the other plot is more sensible in that it uses less ink. Finally, we note that in this case of a blockplot the diagonal is not occupied by a constant but contains instead the fraction of non-missing $(n(x, x)/n)$ and missing $(m(x, x)/n)$ values, respectively, for each individual variable X. The two tables have inverse relationships between the diagonal and off-diagonal elements: $n(x, x) \geq n(x, y)$ and $m(x, x) \leq m(x, y)$. That is, in the $n(x, y)$-table the diagonal dominates its row and column, whereas in the $m(x, y)$-table the diagonal is dominated by its row and column.

3.5 Marginal and Bivariate Plots: Histograms/Barcharts, Scatterplots, and Scatterplot Matrices

The correlation of a pair of variables is a simple summary measure of association between two variables, hence one often wonders about the detailed nature of the association. The full details can be learned from a scatterplot of the two variables. Often the association is constrained by the marginal distribution, hence we also show histograms and barcharts. Figure 7 shows three examples of triples consisting of a pairwise scatterplot and two marginal histograms (for quantitative variables) and barcharts (for categorical variables). From Figure 7 we can draw a few conclusions and recommendations:

- A most basic use of the plots is to note the **type** of the variables: In Figure 7, both variables on the left (`..nonverbal_iq..` and `..verbal_iq..`) and the y-variable in the center (`..vma..`) are **quantitative**, the x-variable in the center (`ados_module..`) is apparently **ordinal** with four levels, and both variables on the right (`nonverbal_iq_type` and `verbal_iq_type`) are **binary**. Quantitative variables can have strong marginal features: It might be of interest to observe that the x-variable on the left is slightly bimodal, with a major mode around $x = 90$ and a minor mode around $x = 30$.[8] The y-variable in the center

[8] The bimodality of the IQ distribution is a measurement artifact: For cognitively highly impaired probands a different and more appropriate IQ test is administered. In theory this alternative test should be

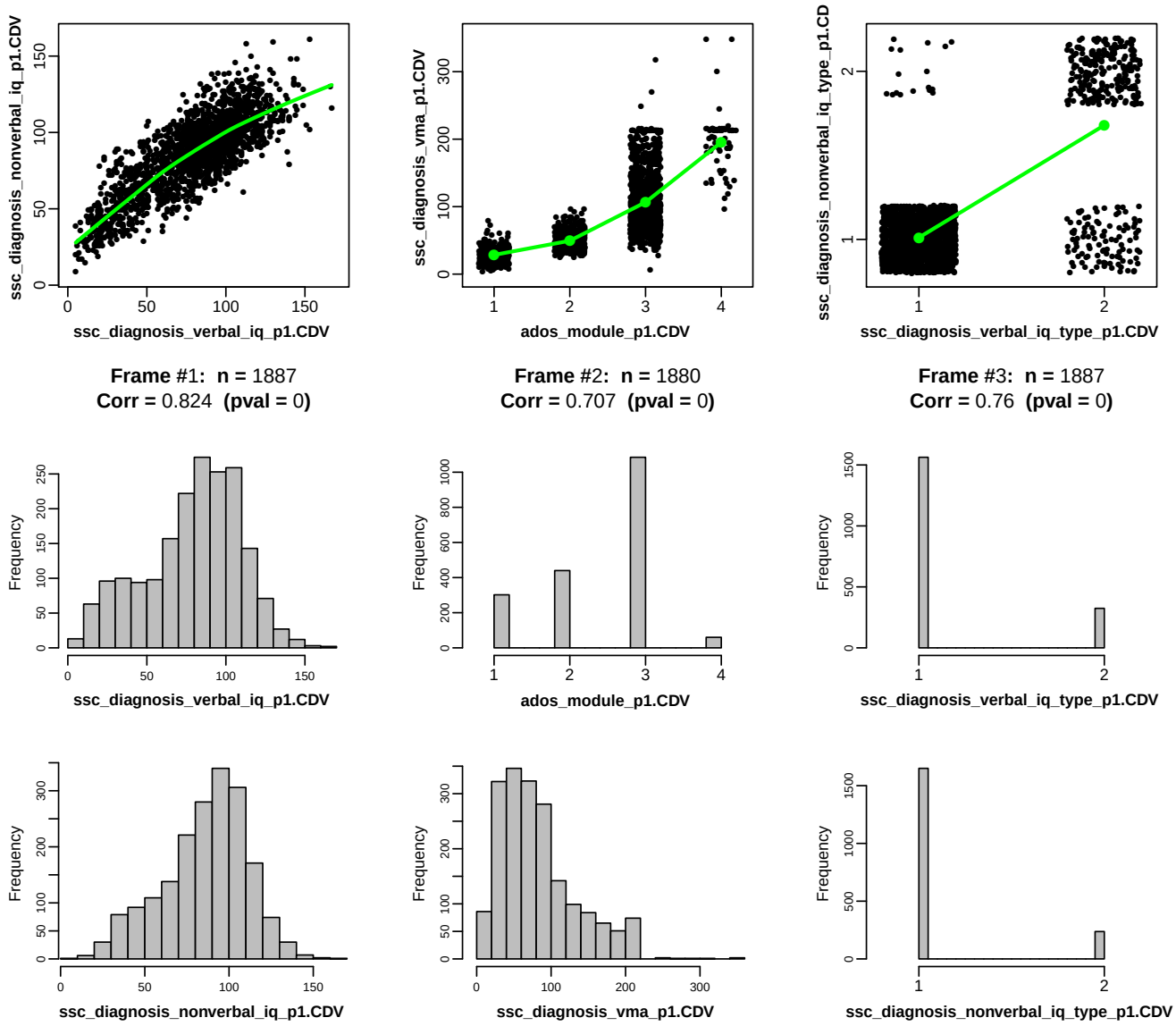

Figure 7: *Scatterplots and histograms/barplots for three variable pairs.*

scatterplot is partially censored on the upper side at about $y = 210$, as can be seen both in the scatterplot and in the (lower) histogram.

- **Categorical variables**, when scored numerically, can be gainfully displayed in scatterplots. It is useful to **jitter** them to avoid being misled by overplotting. In Figure 7, jittering is applied to the x-variable in the center scatterplot and to both binary variables in the right hand scatterplot.

- To enhance the perception of the **association**, the scatterplots can be decorated with **smooths** for continuous variables and with **traces of group means** when the x-variable is categorical with fewer than, say, 8 groups (default, can be changed). In the left and center scatterplots of Figure 7, the associations of the y-variables with the x-variables are seen to be somewhat non-linear, but compared to the linear component of the association, the non-linearities are relatively modest.[9]

The ***AN*** shows scatterplots and histograms/barcharts in a window separate from the blockplot window, one triple of plots at a time. To overcome the one-at-a time limitation, the ***AN*** also offers scatterplot matrices (sometimes called "sploms") of arbitrary numbers of variables. An example, involving four variables (different from those in Figure 7), is shown in Figure 8. For readers not familiar with scatterplot matrices, note that each variable pair is shown twice, in plots located symmetrically off the diagonal, and with reverse roles as x- and y-variables. Each diagonal cell shows a variable label that indicates (1) the common x-axis in the column of the cell and (2) the common y-axis in the row of the cell. For the reader familier with scatterplot matrices, note that we show the vertical order of the variables ascending from bottom to top, the reason being consistency with the convention we use in the blockplots.

As for particulars of the scatterplot matrix shown in Figure 8, the visually most striking features concern marginal distributions, not associations: The first variable is capped at the maximal value +90, and the fourth variable is binary. Otherwise the associations look simply monotone and seem well-summarized by correlations.

3.6 Variations on Blockplots

Blockplots are not the most common visualizations of correlation tables. As a google search of "correlation plot" reveals, the most frequent visual rendering of correlation tables is in terms of "heatmaps" where square cells are always filled and numeric values are coded on a gray or color scale. An example is shown in the left frame of Figure 9; for comparison, the right frame shows the corresponding blockplot. Here are a few observations about the two types of plots:

scaled to cohere with the test administered to the majority, but in practice it creates a minor mode in the low end of the IQ distribution, more so for verbal IQ than nonverbal IQ.

[9] The non-linearity on the left could be due to the marginal distributions. The non-linearity in the center is expected by the expert: verbal mental age (`vma`) on the y-axis should be considerably higher on average in ADOS modules 3 and especially 4 because these modules or levels are formed from a simple test of language competence.

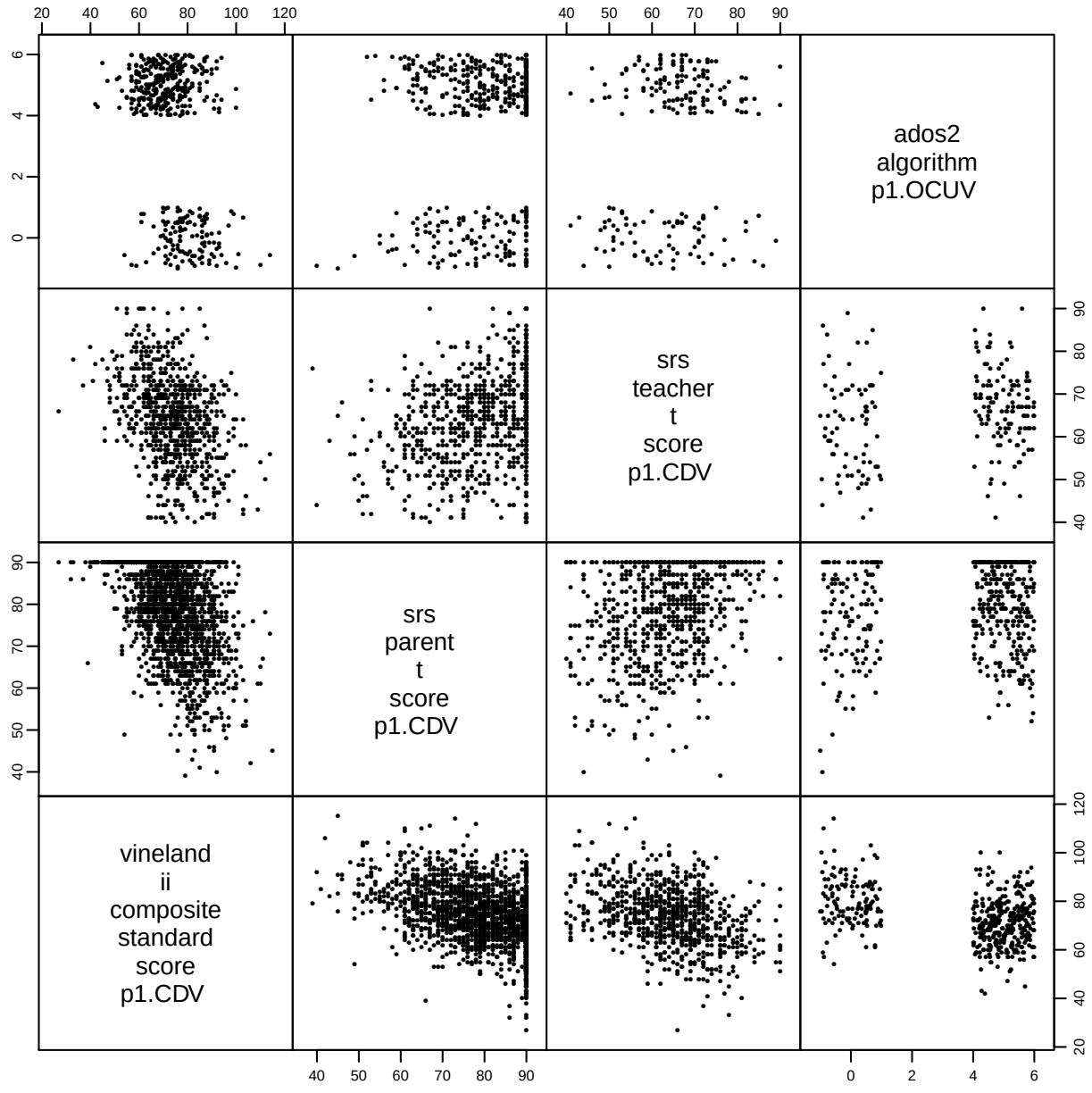

Figure 8: *Scatterplot matrix of four variables. (Note the convention for the vertical order of the variables: bottom to top, for consistency with the blockplots.)*

- Color and gray scale is generally a weaker visual cue than size. This argument favors blockplots as long as the blocks are not too small, that is, as long as the view is not zoomed out too much. The superiority of blockplots over heatmaps is also noted by Wickham, Hofmann and Cook (2006, Figure 2).
- In heatmaps color fuses adjacent cells when they are close in value. This may or may not be a problem for the trained eye, but there is a loss of identity of the rows and columns in heatmaps.
- Heatmaps do not permit mark-up with background color because they fill the square or rectangular cells completely. This problem can be overcome by shrinking the heatmap cells somewhat to allow some surrounding space to be freed up that can be filled with background color for mark-up, as shown in the center frame of Figure 9. This method of rendering, however, seems to further decrease the crispness of heatmaps.
- Heatmaps perform nicely when the view is heavily zoomed out, in which case the individual blocks are so small that size is no longer visually functional as a cue. In this case color coding works well and gives an accurate impression of global structure. We solve this problem for blockplots by showing only 10,000 or so of the largest correlations when heavily zoomed out. Thinning the table in this manner works well even when the visible table is so large that each cell is strictly speaking below the pixel resolution of a raster screen.

Because none of the two types of plots — blockplots or heatmaps — may be uniformly superior at all scales, the ***AN*** provides both, and with one keystroke one can toggle between the two rendering methods. Varying block size allows for the mixed variant shown in the center of Figure 9.

Visualization of correlation tables has a small literature in statistics. An early reference that addresses large correlation tables is Hills (1969), who applies half-normal plots to tell statistically significant from insignificant correlations and clusters variables visually in two-dimensional projections. Closer to the present work are articles by Murdoch and Chow (1996) and Friendly (2002). Both propose relatively complex renderings of correlations with ellipses or augmented circles that may not scale up to the sizes of tables we have in mind but may be useful for conveying richer information for tables that are smaller, yet too large for numeric table display. Blockplot coding, which uses squares, has the advantage that these shapes can completely fill their cells to represent extremal correlations as these are geometrically similar to the shapes of the containing cells (at least if the the default aspect ratio of the blockplot is maintained), whereas all other shapes leave residual space even when maximally expanded.

What we prefer to call descriptively "blockplots", possibly contracted to "blots", has previously been named "fluctuation diagrams" (Hofmann 2000). Under this term one can find a static implementation in the **R**-package `extracat` on the `CRAN` site authored by Pilhoefer and Unwind (2013). Static software for heatmaps is readily available, for example, in the **R**-function `heatmap()`. Heatmaps are often applied to raw data tables, but they can be equally applied to correlation tables. Many variations of glyph coding can be found in the

classic book by Bertin (1983).

An interesting aspect of blockplots is that there exists science regarding the perception of area size. A general theory holds that most continuous stimuli ("continua" such as length, area, volume, weight, brightness, loudness, ...) result in perceptions according to "Stevens' power law" (Stevens (1957), Stevens and Galanter (1957) and Stevens (1975)). That is, a quantitative stimulus x translates to a quantitative perception $p(x)$ through a law of the form $p(x) = c\,x^{\beta}$. As discussed by Cleveland (1985, p. 243) with reference to Stevens (1975), for area perception the power is about $\beta = 0.7$, meaning that an actual area ratio of 2:1 is on average perceived as a ratio of $(2{:}1)^{0.7} \approx 1.62$. This law can be leveraged to determine the transformation that should be used to map correlations to squares in a blockplot. In **R** the symbol size is parametrized in terms of a linear expansion factor called `cex` ('character expansion'). Our goal is to use block sizes such that their perceived ratios faithfully reflect the ratios of respective correlations. This results in the condition $\text{cor} \sim p(\texttt{cex}^2) = (\texttt{cex}^2)^{0.7} = \texttt{cex}^{1.4}$, hence $\texttt{cex} \sim \text{cor}^{1/1.4} \approx \text{cor}^{0.7}$. This is indeed the default power transformation in the ***AN***, although users can change it (see Appendix B). If most correlations are very small, a power closer to zero will expand the range of small values, resulting in enhanced discrimination at the low end at the cost of attenuated discrimination at the high end.

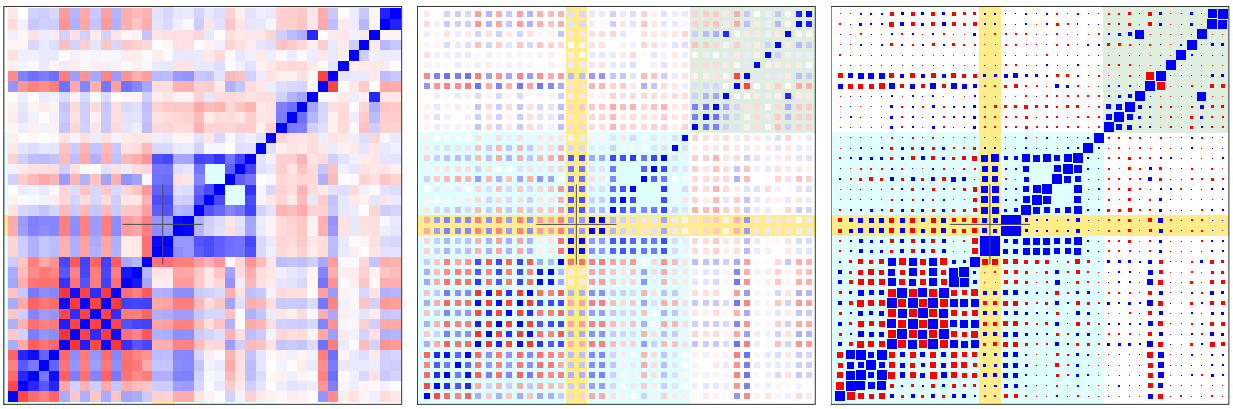

Figure 9: *A heatmap (left) compared with a corresponding blockplot (right), as well as a "shrunk heatmap" in the center.*

4 Operation of the Association Navigator

The purpose of the ***AN*** is to generate the displays described above in rapid order and even with realtime motion. Numerous realtime operations are under mouse and keyboard control, while a few text-based operations are under dialog and menu control. Further parameters can be controlled from the **R** language (see Appendix B), but this will not be necessary for most users. This section describes the operations of the ***AN***, the purposes they serve, as well as a minimal set of **R**-related instructions that concern one-time setup, regular starting up, and saving of state. The software will be available as a **R**-package, but the instructions below do not reflect this and get the reader going by sourcing the software from the first author's site.

4.1 Starting Up the *AN*

In order to simply see some ***AN*** running, the reader may paste the following code into an **R** interpreter:

```
source("http://stat.wharton.upenn.edu/~buja/association-navigator.R")
p <- 200
mymatrix <- matrix(rnorm(20000),ncol=p)
colnames(mymatrix) <- paste("V", 1:p, "_", c(rep("A",p/2),rep("B",p/2)), sep="")
a.n <- a.nav.create(mymatrix)
a.nav.run(a.n)
```

This code will download and source the software, generate an artificial data matrix of normal random numbers, generate an instance of an ***AN*** from it, and start up by creating a window showing a blockplot of correlations as they arise from pure random association among 100 variables given a sample size of 200, divided into two block of 100 variables each, suffixed "A" and "B", respectively. The reader may left-drag the mouse in the plot to see a first realtime response.

To prevent confusion in the operation of an ***AN***, users should note the following fundamental points:

- **Important:** While the ***AN*** is running, the **R** interpreter (**R** Gui) is blocked by the execution of the ***AN***'s event loop! All interactions must be directed at the master window of the ***AN***, which usually shows a blockplot.

- **Quitting the *AN*** and returning to the **R** interpreter is done by typing the capital letter '`Q`' into the ***AN*** master window. The master window will remain as a passive **R** plot window. It will no longer respond to user input, but the **R** interpreter (**R** Gui) will be responsive again. (A live ***AN*** can also be stopped violently by typing interrupt characters `ctrl-C` into the **R** interpreter or by killing the ***AN*** master window, but an educated **R** user wouldn't be this crude.)

- **Help:** On typing the letter '`h`' into a live ***AN***, a help window will appear with terse documentation of all ***AN*** interactions. The window is meant to give reminders to previously initiated ***AN*** users, not introductions to beginners. — The help window is actually a menu such that selecting a line documenting a keystroke will emulate the effects of the keystroke. Because the help window is a menu, it must be closed in order to regain the ***AN***'s attention. (This behavior will be changed in a future version.)

- **Notion of 'state':** An ***AN*** instance has internal state. As a consequence, whenever a user stops a live ***AN*** and restarts it, it will resume in the exact state in which it was stopped.

- **Saving 'state':** From the previous point follows that state of an ***AN*** is saved across **R** sessions if the core image has been saved (`save.image()`) before quitting the **R** sessions.

4.2 Moving Around: Crosshair Placement, Panning and Zooming

When an ***AN*** is run for the first time, it shows an overview of the complete correlation table, which may comprise hundreds of variables. Most likely the variables will be organized in variable groups that are characterized by shared suffixes of variable names and visually form a series of highlight squares along the ascending diagonal. The first order of business is to zoom in and pan up and down the ascending diagonal to gain an overview of these sub-tables. Here are the steps:

- **Crosshair:** Place it by left-clicking anywhere in the plotting area. All subsequent zooming is done with regard to the location of the crosshair; it is also the reference point for some panning operations. Repeat left-clicking a few times for practice. The last location of the crosshair will be the target for zooming, described next.

- **Zooming:** Hit the following for a single step of zooming, or keep depressed for "continuous" zooming.

  - '`i`' for zooming in (alternate: '`=`').
  - '`I`' for accelerated zooming in (alternate: '`+`').
  - '`o`' for zooming out (alternate: '`-`').
  - '`O`' for accelerated zooming out (alternate: '`_`').

  Accelerated zooming changes the visible range by a factor 2, whereas regular zooming is adjusted such that 12 steps change the visible range by a factor of 2. Thus the accelerated zooms are usually done discretely with single keystrokes, and the regular zooms in "continuous" mode with depressed keys. For practice, zoom in and out a few times with your choice of key alternates.

- **Panning** (shifting, translating) is most frequently done by dragging the mouse, but keystrokes are sometimes useful for vertical, horizontal, and diagonal searching.
  - Left-depress the mouse and drag; the plot will follow. When heavily zoomed out from a large table, the response may be slow. The response to mouse dragging will be the swifter the more zoomed in the view is.
  - '←', '→', '↑', '↓' for translation in the obvious directions by one block/variable per keystroke.
  - '`d`'/'`D`' for diagonal moves down/up the ascending 45 degree diagonal.
  - ' ', the space bar for accelerated panning by doing the last single-step keyboard move in jumps of five blocks/variables instead of one.
  - '`.`' to pan so the crosshair location becomes the center of the view.
  - '`[`', '`]`', '`{`', '`}`' to pan so the crosshair location becomes, respectively, the bottom left, the bottom right, the top left, or the top right of the view.

  Yet another method of panning will be described below under "Text Search for Variable Names." Combined pan/zoom based on focus rectangles is described in the next subsection.

4.3 Graphical Parameters

Graphical parameters that determine the aesthetics of a plot are rarely gotten right by automatic algorithms. The problem of aesthetics is particularly difficult when zooming in and out over several orders of magnitude is the order of the day. The ***AN*** therefore makes not even an attempt to guess pleasing and much less optimal values for such graphical parameters as font size of variable labels and margin size in blockplots. Instead, the user gets to choose them by trial and error as follows:

- Block size in the blockplot: hit or depress
  - '`b`' to decrease,
  - '`B`' to increase.

  After starting up a new ***AN***, adjusting the block size is usually the second operation after zooming in.
- Crosshair size: hit or depress
  - '`c`' to decrease,
  - '`C`' to increase.

  Exploding the crosshair by depressing '`C`' is an effective method for reading the variable names of a given block in the margins.
- Font size of the variable labels: hit or depress

  - '`f`' to decrease,
  - '`F`' to increase.

  Important: When the font size is large in relation to the zoom, the variable labels get "thinned out" to avoid gross overplotting (only every second, third ... label might be shown). This allows viewers to at least identify the variable group from the suffix.

- Margin size for the variable labels: hit or depress

  - '`m`' to decrease,
  - '`M`' to increase.

  Margin size needs adjusting according to the prevalent label length and font size. A dilemma occurs when, for example, the x-variable labels are much shorter than the y-variable labels. For this situation we want the following:

- Differential margin size for the variable labels: hit or depress

  - '`n`' to decrease the left/y margin and increase the bottom/x margin,
  - '`N`' to increase the left/y margin and decrease the bottom/x margin.

4.4 Correlations, P-values, Missing and Complete Pairs

By default the blockplot of a ***AN*** represents correlations, but the user can choose them to represent p-values or fraction of missing (incomplete) pairs or fraction of complete pairs as follows: Hit

- '`ctrl-O`' for observed correlations,
- '`ctrl-P`' for p-values of the correlations (Section 3.3),
- '`ctrl-M`' for fraction of missing/incomplete pairs (Section 3.4),
- '`ctrl-N`' for fraction of complete pairs (Section 3.4),.

As discussed in Section 3.3, p-values can be thresholded to obtain Bonferroni-style protection against multiplicity. The thresholds are confined to a ladder of "round" values. Stepping up and down the ladder is achieved by repeatedly hitting

- '`>`' to lower the threshold and obtain greater protection,
- '`<`' to raise the threshold and lose protection.

Recall Figure 5 for two examples of p-value blockplots that differ in the threshold only. — Thresholding also applies to correlation blockplots, in which case '`>`' raises the threshold on the magnitude of the correlations that are shown, and '`<`' lowers it.

Sometimes it is useful to compare magnitudes of the blocks without the distraction of color, hence it may be convenient to hit

- '`ctrl-A`' to toggle between showing all blocks in blue (ignoring signs) and showing the negative correlations (and their p-values) in red.

4.5 Highlighting (1): Strips

Highlight strips are horizontal or vertical bands that run across the whole width or height of the blockplot. They help users search the associations of a given variable with all other variables. Cross-wise highlight strips are also often placed to maintain the connection between a given block and the labels of the associated variable pair. By default the color of highlight strips is "lightgoldenrod1" in **R**. Their appearance is shown in Figure 2. Highlight strips can co-exist in any number and combination, horizontally and vertically. The mechanisms for creating and removing them are as follows:

- Right-click the mouse on
  - a block in the blockplot to place *a horizontal and a vertical* highlight strip through the block;
  - an x-variable label on the horizontal axis to place a *vertical* highlight strip through this variable;
  - a y-variable label on the vertical axis to place a *horizontal* highlight strip through this variable.
- Hit '`ctrl-C`' to clear the strips and start from scratch.

Instead of clicking one can right-depress and drag the mouse across the blockplot with the effect that horizontal and vertical strips are placed across all blocks touched by the drag motion.

Vertical highlight strips lend themselves to convenient searching of associations between a fixed variable on the horizontal axis and all variables on the vertical axis. To this end it is useful to pan vertically with '↑', '↓', and the space bar as accelerator (Section 4.2).

4.6 Highlighting (2): Rectangles

A highlight rectangle is a rectangular area in the blockplot selected by the user for highlighting. Highlight rectangles are meant to help the user focus on the associations between contiguous groups of variables on the horizontal and the vertical axis. By default the color of highlight rectangles is "lightcyan1" in **R**. Their appearance is that of the center square in Figure 4. In the case of this figure, the highlight rectangle coincides with the highlight square for the variable group defined by the suffix "`p1.CDV`". Unlike highlight squares, which mark predefined variable groups, highlight rectangles can be placed (and removed from) anywhere by the user. The mechanisms to this end are as follows:

- Define a highlight rectangle in arbitrary position by placing two opposite corners:
  - Place the crosshair in the location of the desired first corner; then hit '`1`' to place the first corner of a new rectangle.
  - Place the crosshair in the location of the desired second corner; then hit '`2`' to place the second corner.

Action '1' creates a new highlight rectangle consisting of just one block. Action '2' never creates a new block but only sets/resets the second corner of the most recent rectangle.

- Define a highlight rectangle in terms of two variable groups:
  - Place the crosshair such that the x-coordinate is in the desired horizontal variable group and the y-coordinate in the desired vertical variable group; then
  - hit '3' to create the highlight rectangle.

  As a special case, this allows a highlight square to become a highlight rectangle by letting the x- and y-variable groups be the same, as in Figure 4.

- Pan and zoom to snap the view and the highlight rectangle to each other:
  - Place the crosshair in the highlight rectangle to be snapped; then
  - either hit '4' to snap, preserving the aspect ratio,
  - or hit '5' to snap, distorting the aspect ratio, unless the rectangle is a square.

  If the crosshair is not placed in a highlight rectangle, the most recent one will be used. Note that the squares in a blockplot always remain squares, even if the aspect ratio of the plot has been distorted. Changing the aspect ratio has the consequence that the squares can no longer fill their cells because they have become rectangles.

- Any number of highlight rectangles can co-exist. Remove them selectively as follows:
  - Place the crosshair anywhere in a highlight rectangle to be removed; then
  - hit '0' to remove it.

4.7 Reference variables

A recurrent issue when using the ***AN*** is that some variables are often of persistent interest. In autism phenotype data, for example, a recurrent theme is to check up on age, gender and site association (potential confounders) while examining associations within and between various "autism instruments" such as ADOS, ADI, RBS,... To spare users the distraction of hopping back and forth across the multi-hundred square table, the ***AN*** implements a notion of "reference variables", that is, variables that never disappear from view. The ***AN*** keeps them tucked in the left and the bottom of the blockplot. The manner in which reference variables present themselves is shown in Figure 10. The mechanism for selecting reference variables is by first selecting them with highlight strips (Section 4.5), and then hitting

- 'R' to turn the strip variables into reference variables,
- 'r' to toggle on and off the display of the selected reference variables.

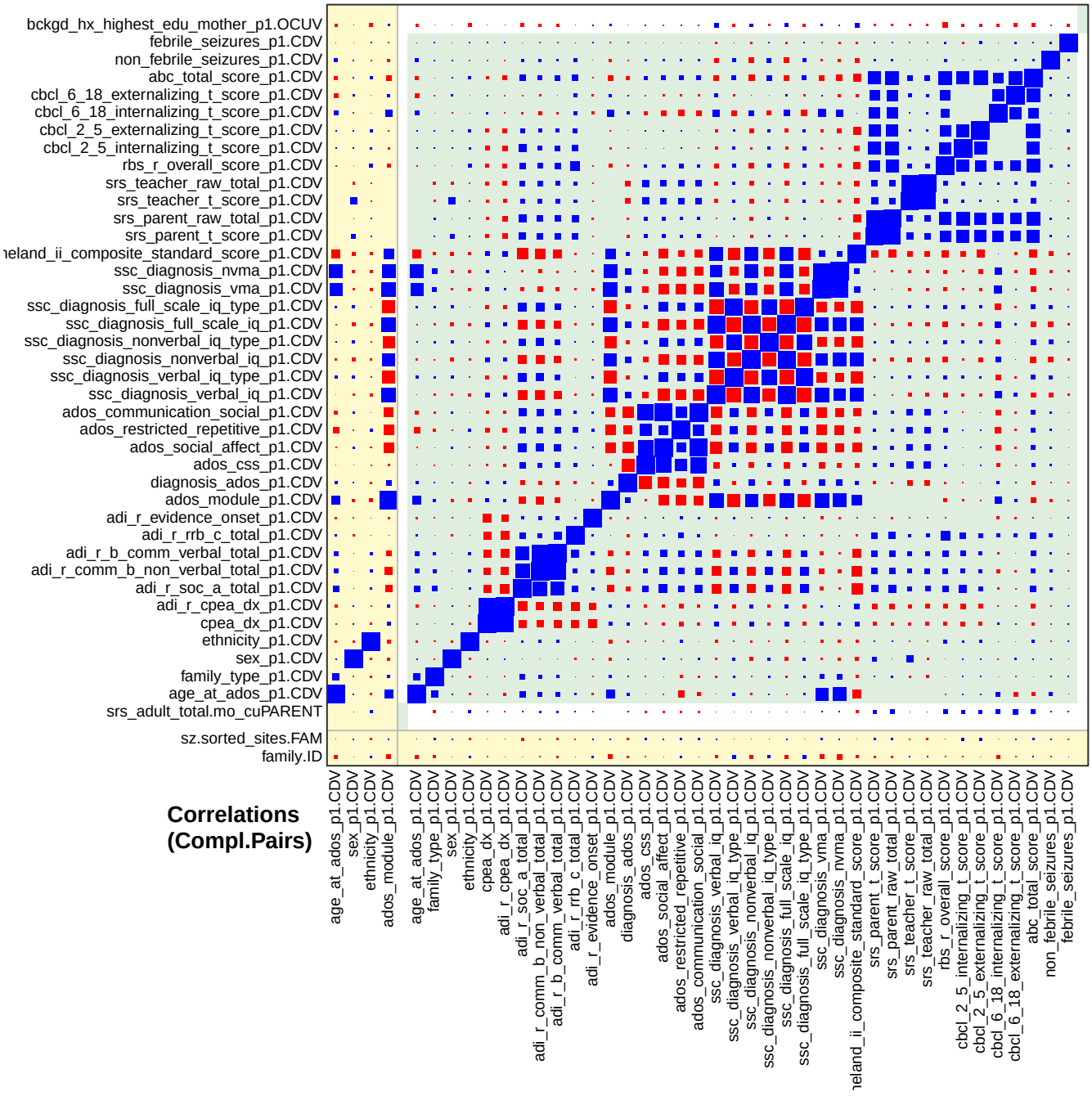

Figure 10: *Reference variables shown in the left and bottom bands. Whenever the user zooms and pans the blockplot, these variables stay in place and show their associations with the variables from the rest of the blockplot.*

The disentangling of the two actions allows users to keep marking up strips without changing the earlier selected reference variables.

In Figure 10, the y-reference variables are "`sz.sorted_sites.FAM`" and "`family.ID`", and their associations with the x-variables are shown in the horizontal band at the bottom. Similarly, the x-reference variables are "`age_a_ados_p1.CDV`", "`sex_p1.CDV`", "`ethnicity_p1.CDV`" and "`ados_module_p1.CDV`", and their associations with the y-variables are shown in the vertical band on the left. In the bottom left corner, the intersection of the reference bands, are shown the associations between x- and y reference variables.

4.8 Searching Variables

Other recurrent issues with analyzing large numbers of variables is simply finding variables. For example,

- find a variable whose name one remembers partly, but not exactly; or
- find a set of variables whose names share a meaningful syllable.

In the context of autism, for example, it might be of interest to find all variables related to anxiety across all instruments; it would then be sensible to search for all variables that contain the phoneme "`anx`" in their name. This type of problem can be solved in the ***AN*** with a blend of text search and menu selection. We address here the problem of locating one variable and panning to it. To this end hit...

- '`H`' to locate a variable on the x-axis;
- '`V`' to locate a variable on the y-axis;
- '`@`' to locate a variable on both the x- and the y-axis.

In each case a dialog box pops up where a search string or regular expression can be entered. On hitting '`<Return>`' or '`OK`', a menu appears with the list of variables that contains the search string or matches the regular expression (according to **R**'s `grep()` function). The user is then asked to select one of the offered variables, upon which the ***AN*** pans to the variable (depending on '`H`', '`V`' or '`@`') on the x- or the y-axis or both, marks it with a vertical or horizontal highlight strip or both, and places the crosshair on it. See Figure 11.

Search can be bypassed by not entering a search string at all. The menu shows then the complete list of all variables with scrolling.

4.9 Lenses: Scatterplots and Barplots/Histograms

We think of barplots, histograms and scatterplots as lenses into the blocks, each of which represents a pair (x, y) of variables. Taking the pair "under the lens" means looking at the association (and the marginal distribution) in greater detail; see Section 3.5 above. The mechanics are as follows: Hit

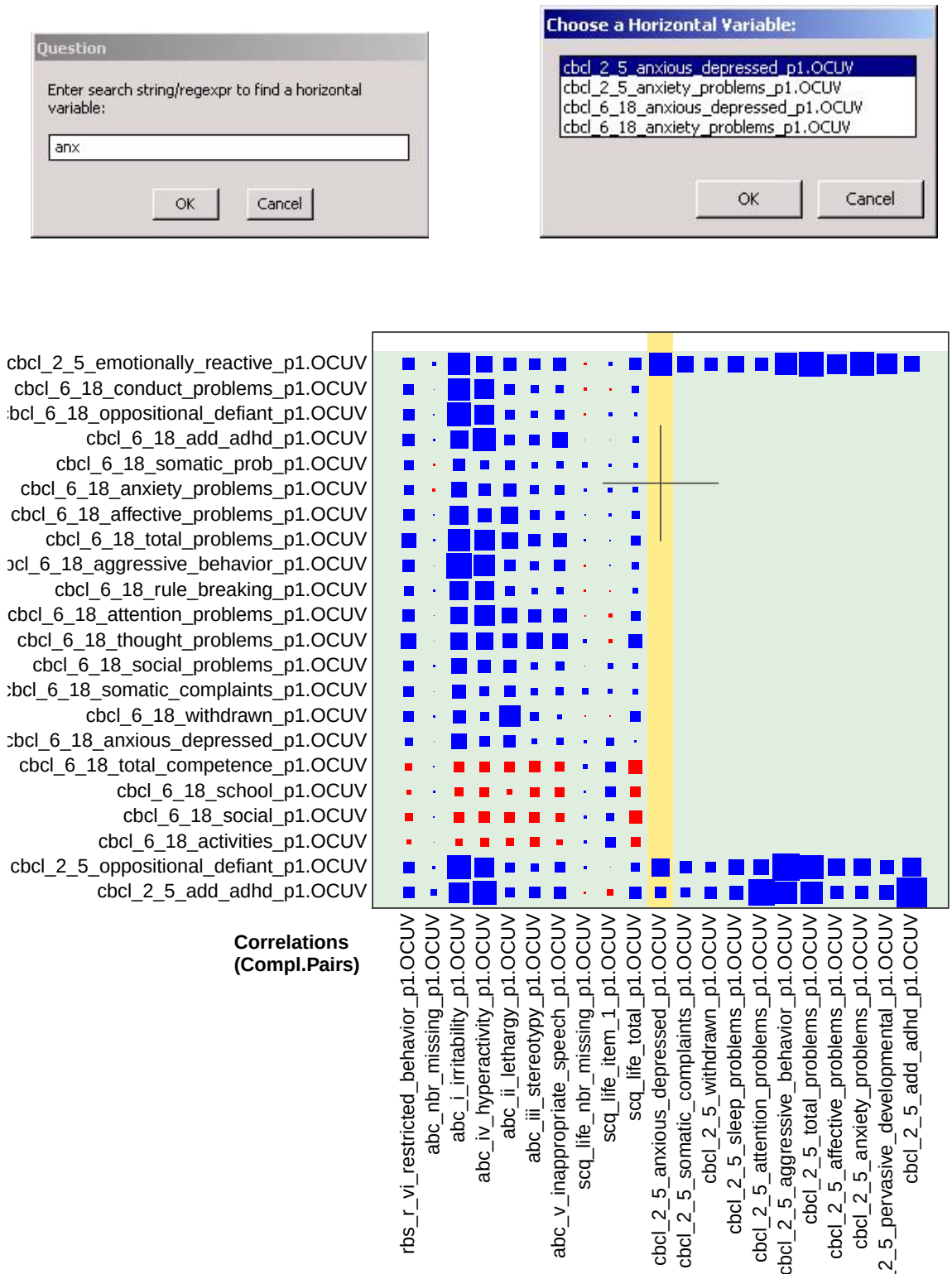

Figure 11: *Text search with 'H' for horizontal variables containing "anx", followed by selection of "cbcl_2_5_anxious_depressed_p1.OCUV". The view pans horizontally to the selected variable, marks it with a vertical hilight strip, and places the crosshair on it.*

- '`x`' to see in a separate window (Figure 7) a scatterplot and barplots/histograms of the two variables marked by the crosshair cursor.
- '`y`' to switch the x-y roles of the variables.
- '`l`' to toggle showing a "line", that is, a smooth if x is quantitative, and a trace of y-means of the x-groups if x is categorical.

Important: The lens window is passive and does not accept interactive input. One must expose the blockplot master window to continue with ***AN*** interactions.

These lenses have a simple history mechanism in that the consecutive x-y variable names are collected in a list that can be traversed and edited: Hit

- '`PgUp`' to take one step back in the history,
- '`PgDn`' to take one step forward in the history,
- '`Home`' to jump to the beginning of the history,
- '`End`' to jump to the end of the history (the present),
- '`Delete`' to delete the current lens from the history.

Finally, there is a separate lens mechanism with its own window that shows all pairwise scatterplots of the variables currently in highlight strips. An example is shown in Figure 8. As to the mechanics, hit

- '`z`' to create the scatterplot matrix with independently scaled axes;
- '`Z`' to create the scatterplot matrix with identically scaled axes.

The latter option is sometimes useful when all variables live on the same scale but have somewhat different ranges.

4.10 Color-Brushing in Scatterplots

Often one would like to focus on groups of cases in the scatterplots of the lens window. This can be achieved with color brushing as follows:

- Hit '`s`' to see the current lens scatterplot in the main window, replacing the blockplot.
- Hit '`r`' to fix one corner of a brush at the current mouse location.
- Left-depress and drag the mouse: the rectangular brushing area should open up and change shape. Whenever the brush moves over a scatterplot point, it will change color.
- Right-depress and drag the mouse: the rectangular brushing area will translate along with the mouse. Again, moving over scatterplot points will change their color.
- The brushing color can be changed by cycling through a series of colors, hitting '`S`'. The color gray does not paint; it is useful for counting the points under the brush as their number is shown in the bottom left corner.

Figure 12: *Screenshot of the adjustment menu. As shown, it enables adjustment of the "*`srs`*" variables for "*`age_at_ados_p1.CDV`*" and "*`sex_p1.CDV`*".*

- Hit '`s`' to return to the blockplot in the main window.

Thus, hitting '`S`' toggles between blockplot and scatterplot in the main window. After each brushing operation, the lens scatterplot will follow suit and color its points to match those in the main window.

4.11 Linear Adjustment

Another recurrent task in large tables is what we may call "adjustment". The phrase "adjusting for x" has many synonyms: "accounting for x", "controlling for x", "correcting for x", "allowing for x", and "holding x fixed" or "conditioning on x". Technically most correct is the last expression: We are often interested in the conditional association between variables y and z given (holding fixed) a variable x, as measured for example by the conditional correlation $r(y, z|x)$. In the context of the autism phenotype, one may be interested in adjusting for age and/or gender. In practice, particulary in large-p problems, there is rarely sufficient data to truly estimate conditional distributions,[10] hence one makes the simplifying assumption that all associations are linear with constant conditional variances (homoscedasticity).[11] In that case, adjustment of y for x amounts to a linear regression and forming residuals, that is, "residualizing" or "partialling out" is done by subtracting the equation fitted with linear regression: $y_{\bullet x} = y - (b_0 + b_1 x)$. As a consequence, $r(y_{\bullet x}, x) = 0$, that is, by forming $y_{\bullet x}$ one removes from y the linear association with x. This type of linear adjustment generalizes to multiple x variables by residualizing with regard to a multiple linear regression.

In the ***AN*** implementation of linear adjustment, one has to select a set of "independent" x-variables, called "adjustors", and a set of "dependent" y-variables, called the "adjustees". Often the set of adjustors is small, possibly just one variable such as age, whereas the set of adjustees can be large, for example, all items and summary scales of an autism phenotype instrument such as the `SRS` ("Social Responsiveness Scale"). The selection mechanisms are the same for both adjustors and adjustees: text search or regular expression matching, followed by menu selection, similar to Section 4.8, but here the menu selection allows multiple choices. The mechanics are as follows: Hit

- '`A`' to call up a large menu that forms the interface for all adjustment operations.

An example is shown in Figure 12. Initially, the list of adjustors and adjustees will be empty, so both need to be populated with text searches that require a dialog initiated by selecting the lines "`Find ADJUSTORS...`" and "`Find ADJUSTEES...`" in sequence. Figure 12 shows the state after having matched the regular expression "`age_at|sex_p1.CDV`" for adjustors and searched the string "`CDV`" for adjustees.

Finally, after selection of adjustors and adjustees is completed, the user may select the top line of the menu to actually "`Do Adjustment`". Each raw adjustee will then be replaced

[10] Natural exceptions do exist: If we analyze females and males separate, for example, we study gender-conditional associations.

[11] Both assumptions may be wrong, but some form of adjustment, even if flawed, is often more informative than remaining with raw variables.

by its residuals obtained from the regression onto the adjustors. (To undo adjustment, select the second line from the Adjustment dialog, "`Undo Adjustment`".)

To assist the visual examination of adjustment results, one may want to select the third line from the top of the menu in order to highlight the adjustors among the x-variables and the adjustees among the y-variables ("`Mark with Highlight Strips...`"). Turning them further into reference variables (Section 4.7) by hitting "`R`", we obtain Figure 13. As it should be, the correlations between the two adjustors on the x-axis and the many adjustees on the y-axis vanish. The correlations of the adjustees with other variables many now be of renewed interest because they are free of age and gender "effects", which would invite a search of the correlations in the horizontal band of the adjustees.

A word of caution: Adjustment of a y-variable is done using only cases for which there are no missing values among the adjustors and obviously the adjustee is not missing either. Thus the underlying set of cases may have been inadvertently decreased. It is therefore good advice to check the missing-pairs patterns with either '`ctrl-M`' or '`ctrl-N`' (Section 4.4) or by looking at scatterplots (Section 4.9).

Having done adjustment of variables, one often wonders how much of it was done and to which variable. To answer this question, select the fourth line from the adjustment dialog ("`Sort Adjustees...`"): The result is a sorted list of the adjustees according to the R^2 values from the regression of the adjustees/y-variables onto the adjustors/x-variables. See Figure 14 for an example.

4.12 The Future of the *AN*

The functionality described here reflects the 2015 implementation of the ***AN***. Changes to the are planned, the major one being a redesign to give the lens windows interactive responsiveness as well. Currently all interaction is funneled throught the blockplot window, even if the actions affect the lens window.

Other obvious functionality is still missing, above all sorting of variables, manual and algorithmic, and a limited set of sorting operations may be added in a future version of the ***AN***. If readers of this document and users of the ***AN*** have further suggestions, the authors would appreciate hearing.

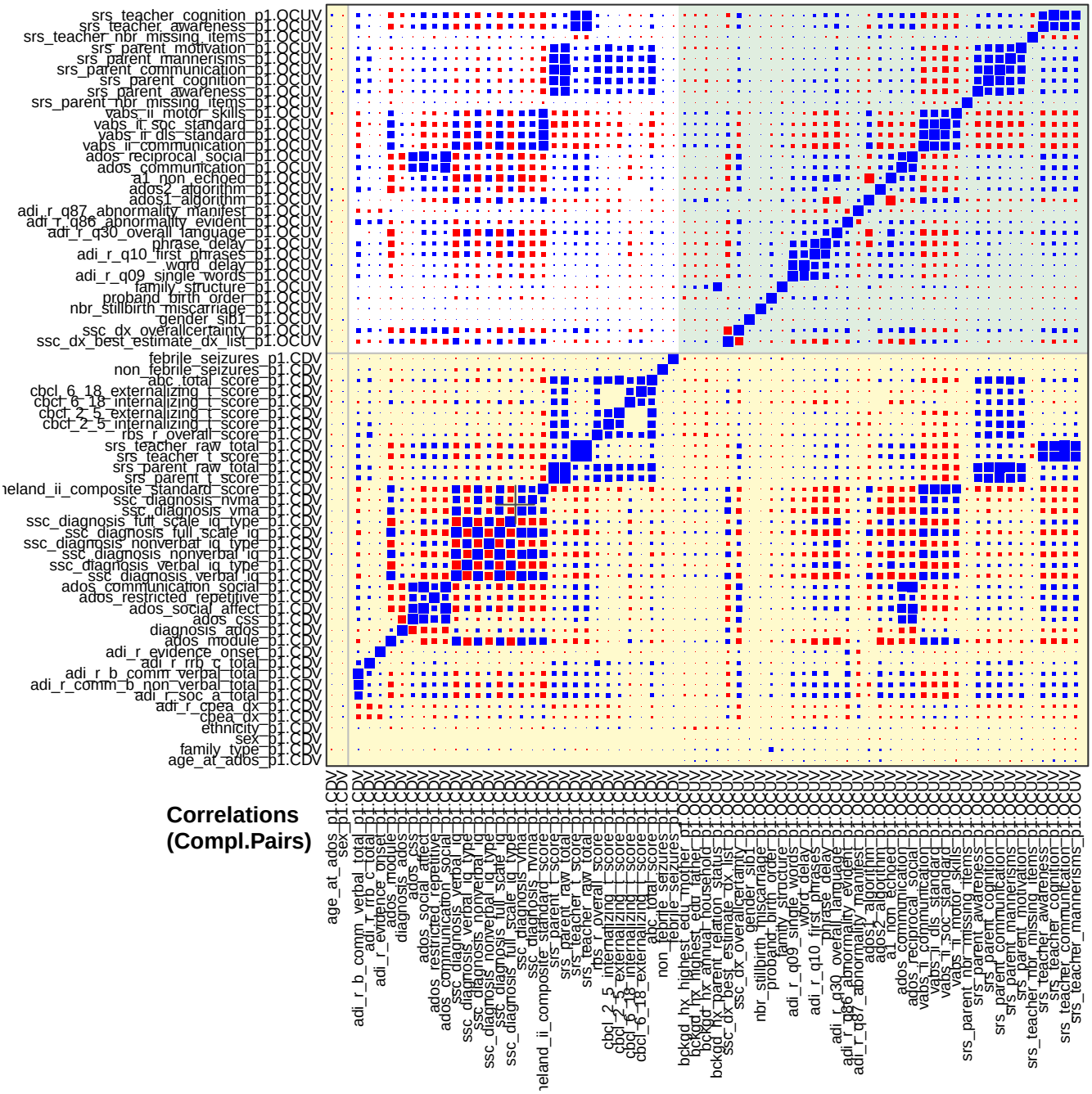

Figure 13: *Results of adjustment of the "*`CDV`*" variables for "*`age_at_ados_p1.CDV`*" and "*`sex_p1.CDV`*": the former are reference variables on the y-axis, the latter on the x-axis. As it should be, the correlations between adjustors and adjustees vanish.*

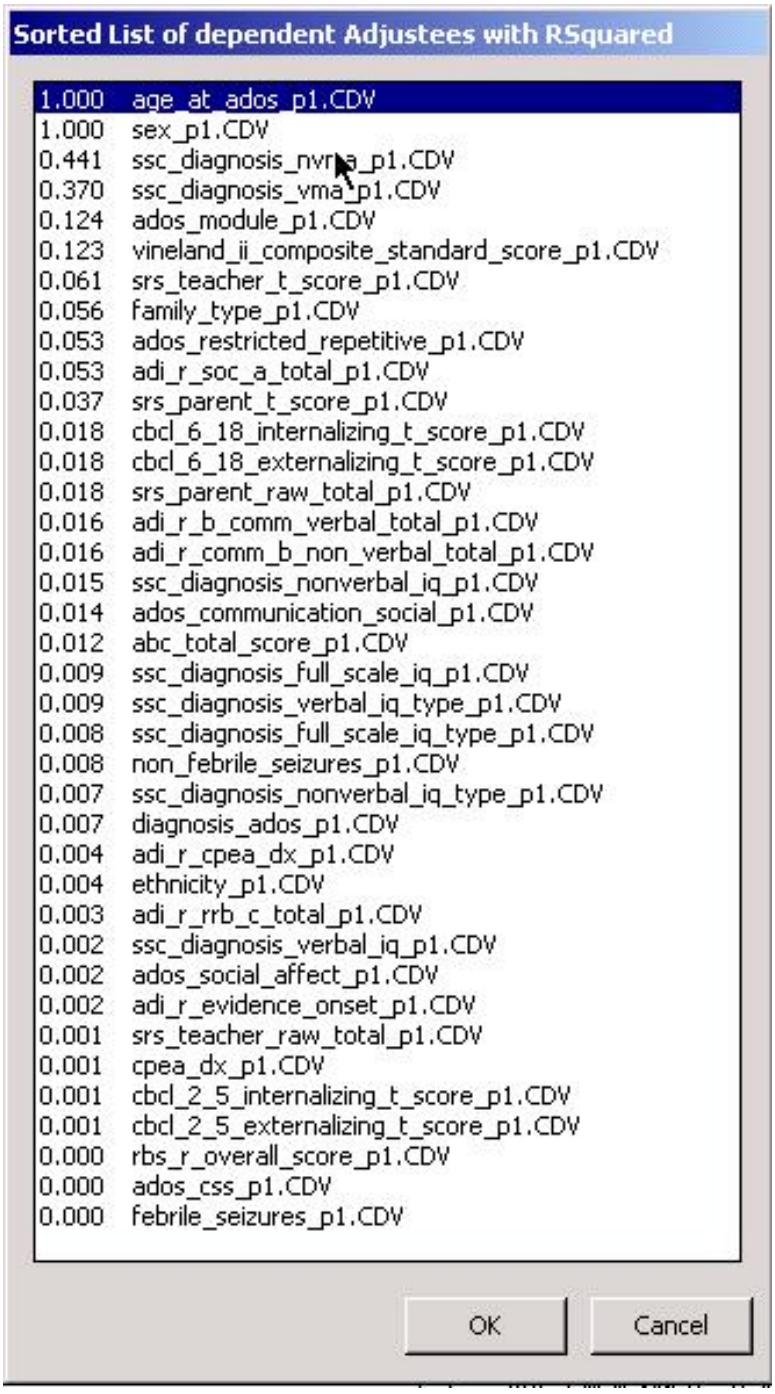

Figure 14: *List of adjustees/y-variables sorted according to the R^2 values from the regressions onto the adjustors/x-variables.*

A Appendix: The Versatility of Correlation Analysis

We return to the apparent limitations of correlations as measures of association which was left as a loose end in the Introduction. We address the objections that (1) correlations are measures of linear association only, (2) correlations reflect bivariate association only, and (3) correlations apply to quantitative variables only. Towards this end we make the following observations and recommendations:

(1) While it is true that correlation is strictly speaking a measure of linear association among quantitative variables, it is also a fact that correlation is useful as a measure of monotone association in general, even when it is non-linear. As long as the association is roughly monotone, correlation will be positive when the association is increasing and negative when it is decreasing. Admittedly, correlation is not an optimal measure of non-linear monotone association, but it is still a useful one, in particular in the large-p problem. Lastly, if gross non-linearity is discovered, it is always possible to replace a variable X with a non-linear transform $f(X)$ (often $\log(X)$) so its association with other variables becomes more linear.[12]

(2) The objection that correlations only reflect bivariate association is factually correct but practically not very relevant. In practical data analysis it is too contrived to entertain the possibility that, for example, there exists association among three variables but there exists no monotone association among each pair of variables.[13] In general one follows the principle that lower-order association is more likely than higher-order association, hence pairwise association is more likely than true interaction among three variables. Therefore data analysts look first for groups of variables that are linked by pairwise association, and thereafter they may examine whether these variables *also* exhibit higher-order association. Note, however, that even multivariate methods such as principal components analysis (PCA) do not detect true higher-order interaction because they, too, rely on correlations only. Finally, we are not asserting that simple correlation analysis should be the end of data analysis, but it should certainly be near the beginning in the large-p problems envisioned here, namely, in the analysis of relatively noisy data as they arise in many social science and medical contexts.[14]

[12] Linearity of association is not a simple concept. For one thing, it is asymmetric: if Y is linearly associated with X, it does not follow that X is linearly associated with Y. The reason is that the definition of linear association, $\mathrm{E}[Y|X] = \beta_0 + \beta_1 X$, is not symmetric in X and Y. Linearity of association in both directions holds only for certain "nice" distributions such as bivariate Gaussians. A counter-exampls is as follows: Let X be *uniformly* distributed on an interval and $Y = \beta_0 + \beta_1 X + \epsilon$ with independent Gaussian ϵ, then Y is linearly associated with X by construction, yet X is *not* linearly associated with Y.

[13] An example would be three variables jointly uniformly distributed on the surface of a 2-sphere in 3-space.

[14] In other large-p problems the variables may be so highly structured that they become intrinsically low-dimensional, as for example in the analysis of libraries of registered images where each variable corresponds to a pixel location and its values consist of intensities at that location across the images. The problem here is not to locate groups of variables with association but to describe the manifold formed by the images in very high-dimensional pixel space. A sensible approach in this case would be non-linear dimension reduction.

(3) The final objection we consider is that correlations do not apply to categorical variables. This objection can be refuted with very practical advice on how to make categorical data quantitative and how to interpret the meaning of the resulting correlations. We discuss several cases in turn:

- If a categorical variable X is **ordinal** (its categories have a natural order), it is common practice to simply number the categories in order and use the resulting integer variable as a quantitative variable. The resulting correlations will be able to reflect monotone association with other variables that may be expressed by saying "the higher categories of X tend to be associated with higher/lower values/categories of other variables." — An obvious objection is that the equi-spaced integers may not be a good quantification of the categories. If this is a serious concern worth some effort, one may want to look into optimal scoring procedures (see, for example, De Leeuw and Rijckevorsel (1980) and Gifi (1990)). The idea behind these methods is to estimate new scores for the categorical variables by making them as linearly associated as possible through optimization of the fit of a joint PCA.
- If a categorical variable X is **binary**, it is common practice to numerically code its two categories with the values 0 and 1, thereby creating a so-called "dummy variable." This practice is pervasive in the Analysis of Variance (ANOVA), but its usefulness is lesser known in multivariate analysis which is our concern. The interpretation of correlations with dummy variables is highly interesting as it solves two seemingly different association problems:
  - First order association between a binary variable X and a quantitative variable Y means that there exists a difference between the two means of Y in the two groups denoted by X. As it turns out, the correlation of a dummy variable X with a quantitative variable Y is mathematically equivalent to forming the t-statistic for a two-sample comparison of the two means of Y in the two categories of X ($t \propto r/(1-r^2)^{1/2}$). Even more, the statistical test for a zero correlation is virtually identical to the t-test for equality of the two means. Thus two-sample mean-comparisons can be subsumed under correlation analysis.
  - Association between two binary variables means that their 2×2 table shows dependence. This situation is usually addressed with Fisher's exact test of independence. It turns out, however, that Fisher's exact test is equivalent to testing the correlation between the dummy variables, the only discrepancy being that the normal approximation used to calculate the p-value of a correlation is just that, an approximation, although an adequate one in most cases.
- If a categorical variable X is truly **nominal** with more than two values, that is, neither binary nor ordinal, we may again follow the lead of ANOVA and replace X with a collection of dummy variables, one per category. For example, if in a

medical context data are collected in multiple sites, it will be of interest to see whether substantive variables in some sites are systematically different from other sites. It is then useful to introduce dummy variables for the sites and examine their correlations with the substantive variables. A significant correlation indicates a significant mean difference at that site compared to the other sites.

This discussion shows that categorical variables can be fruitfully included in correlation analysis, either with numerical coding of ordinal variables, or with dummy coding of binary and nominal variables.

This concludes our discussion of the versatility of correlation analysis.

B Appendix: Creating and Programming *AN* Instances

To create a new instance of an **Association Navigator** for a given dataset, use the following **R** statement:

```
a.n <- a.nav.create(datamatrix)
```

where '`datamatrix`' is a numeric matrix, not a dataframe. The new ***AN*** instance '`a.n`' can be run with the following **R** statement:

```
a.nav.run(a.n)
```

These steps are completely general and may be useful for arbitrary numeric data matrices with up to about 2,000 variables.

Table 1 shows a template for forming potentially useful instances of ***AN***s that display large numbers of SSC phenotype variables. As written the statement would produce an ***AN*** in the order of 3,000 variables.

AN's are implemented not as lists but as "environments," a relatively little known data structure among most **R** users. Environments have some interesting properties. One can look inside an ***AN*** with the **R** idiom

```
with(a.n, objects())
```

in order to list the ***AN***-internal variables inside the ***AN*** instance '`a.n`'. Assignments and any other kind of programming of the internal state variables can be achieved the same way. For example, if one desires a change of color of highlight strips to "mistyrose", one can achieve this with the following:

```
with(a.n, { strips.col <- "mistyrose";  a.nav.blockplot() })
```

The call to `a.nav.blockplot()` redisplays the blockplot with the new paramater setting. Changing the blockplot glyph from square to diamond is achieved with

```
with(a.n, { blot.pch <- 18;  a.nav.blockplot() })
```

and reversing the color convention from "blue = positive" to "red = positive" in the style of heatmaps is done with

```
with(a.n, { blot.col.pos <- 2;  blot.col.neg <- 4;  a.nav.blockplot() })
```

Note, however, that this affects only blockplots, not heatmaps, the latter requiring computation of a color scale, not just a binary color decision. Still, there is plenty of opportunity for playfulness by experimenting with display parameters. A more sophisticated example concerns changing the power transformation that maps correlations to glyph sizes:

```
with(a.n, { blot.pow <- .7; a.nav.cors.trans(); a.nav.blockplot() })
```

In addition to redisplay with `a.nav.blockplot()`, this also requires recomputation of the display table with `a.nav.cors.trans()`.

R environments represent one of the two data types (the other being "connections") that disobeys the functional programming paradigm that is otherwise fundamental to **R**. As a consequence, assignment of an ***AN*** does not allocate a new copy but passes a reference instead. In particular, the **R** statement

```
b.n <- a.n
```

creates a variable '`b.n`' that will be a reference to the same environment as the variable '`a.n`'. Hence the two statements

```
a.nav.run(a.n)
a.nav.run(b.n)
```

will run off the same ***AN*** instance. They have identical effects in the sense that interactive operations affect the same instance.

References

[1] J. Bertin. *Semiology of Graphics*. Madison, WI: The University of Wisconsin Press, 1983.

[2] W. S. Cleveland. *The Elements of Graphing Data*. Pacific Grove, CA: Wadsworth & Brooks/Cole, 1985.

[3] J. De Leeuw and J. van Rijckevorsel. HOMALS and PRINCALS - some generalizations of principal components analysis. In E. Diday et al., editor, *Data Analysis and Informatics II*, pages 231–242. Amsterdam: Elsevier Science Publisher, North Holland, 1980.

[4] M. Friendly. Corrgrams: Exploratory displays of correlation matrices. *The American Statistician*, 56(4):316–324, 2002.

[5] A. Gifi. *Nonlinear multivariate analysis*. New York: John Wiley & Sons, 1990.

[6] M. Hills. On looking at large correlation matrices. *Biometrika*, 56(2):249–253, 1969.

[7] H. Hofmann. Exploring categorical data: Interactive mosaic plots. *Metrika*, 51(1):11–26, 2000.

[8] D. J. Murdoch and E. D. Chow. A graphical display of large correlation matrices. *The American Statistician*, 50(2):178–180, 1996.

[9] A. Pilhoefer and A. Unwin. New approaches in visualization of categorical data: R package extracat. *Journal of Statistical Software*, 53(7):1–25, 2013.

[10] S. S. Stevens. On the psychophysical law. *The Psychological Review*, 64(3):153–181, 1957.

[11] S. S. Stevens. *Psychophysics*. New York: John Wiley & Sons, 1975.

[12] S. S. Stevens and E. H. Galanter. Ratio scales and category scales for a dozen perceptual continua. *Journal of Experimental Psychology*, 54(6):377–411, 1957.

[13] H. Wickham, H. Hofmann, and D. Cook. Exploring cluster analysis. *http://www.had.co.nz/model-vis/clusters.pdf*, 2006.

```
a.n <- a.nav.create(cbind(
    "family.ID"=as.numeric(v.families),
    v.sites,  v.srs.bg,  v.individual,
    v.family,  v.parent.race,  v.parent.common,
    v.proband.cdv, v.proband.ocuv,  v.sibling.s1, v.sibling.s2,
    v.ados.common,
    v.ados.1, v.ados.1.raw, v.ados.2, v.ados.2.raw,
    v.ados.3, v.ados.3.raw, v.ados.4, v.ados.4.raw,
    v.adi.r.diagnostic, v.adi.r.pca, v.adi.r,
    v.adi.r.dum, v.adi.r.loss,
    v.ssc.diagnosis,
    v.vineland.ii.p1, v.vineland.ii.s1,
    v.cbcl.2.5.p1,  v.cbcl.2.5.s1,
    v.cbcl.6.18.p1, v.cbcl.6.18.s1,
    v.abc, v.abc.raw,  v.rbs.r, v.rbs.r.raw,
    v.srs.parent.p1,   v.srs.parent.recode.p1,
    v.srs.teacher.p1,  v.srs.teacher.recode.p1,
    v.srs.parent.s1,   v.srs.parent.recode.s1,
    v.srs.teacher.s1,  v.srs.teacher.recode.s1,
    v.srs.adult.fa,    v.srs.adult.recode.fa,
    v.srs.adult.mo,    v.srs.adult.recode.mo,
    v.bapq.fa, v.bapq.recode.fa, v.bapq.mo, v.bapq.recode.mo,
    v.fhi.interviewer.fa, v.fhi.interviewer.mo,
    v.scq.current.p1, v.scq.life.p1,
    v.scq.current.s1, v.scq.life.s1,
    v.ctopp.nr,  v.purdue.pegboard,  v.dcdq,  v.ppvt,
    v.das.ii.early.years, v.das.ii.school.age,
    v.ctrf.2.5, v.trf.6.18,
    v.ssc.med.hx.v2.autoimmune.disorders, v.ssc.med.hx.v2.birth.defects,
    v.ssc.med.hx.v2.chronic.illnesses,    v.ssc.med.hx.v2.diet.medication.sleep,
    v.ssc.med.hx.v2.genetic.disorders,    v.ssc.med.hx.v2.labor.delivery.birth.feeding,
    v.ssc.med.hx.v2.language.disorders,
    v.ssc.med.hx.v2.medical.history.child.1, v.ssc.med.hx.v2.medical.history.child.2,
    v.ssc.med.hx.v2.medical.history.child.3,
    v.ssc.med.hx.v2.medications.drugs.mother,
    v.ssc.med.hx.v2.neurological.conditions,
    v.ssc.med.hx.v2.other.developmental.disorders, v.ssc.med.hx.v2.pdd,
    v.ssc.med.hx.v2.pregnancy.history, v.ssc.med.hx.v2.pregnancy.illness.vaccinations,
    v.ssc.psuh.fa,     vv.ssc.psuh.mo,
    v.temperature.form.raw
    ), remove=T )
```

Table 1: *Template for joining large numbers of SSC tables and creating an **AN** for them. Readers should make a selection from this template as the full collection creates a data matrix with about 3,000 variables.*