

A Tool for Mining Large Correlation Tables: The Association Navigator

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

November 15, 2010

Abstract

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 $2,000 \times 2,000$. 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.

1 Introduction

This document describes the **Association Navigator** in three sections: (1) In this introductory section we give some background about the data analytic and statistical problem addressed by this tool; (2) in the next section we describe the graphical displays used by the tool; (3) in the final section we describe the actual operation of the tool. We start with some background:

*This work was supported by a grant from the Simons Foundation (SFARI award #121221 to A.K.). We appreciate obtaining access to the phenotypic data on SFARI Base (<https://base.sfari.org>).

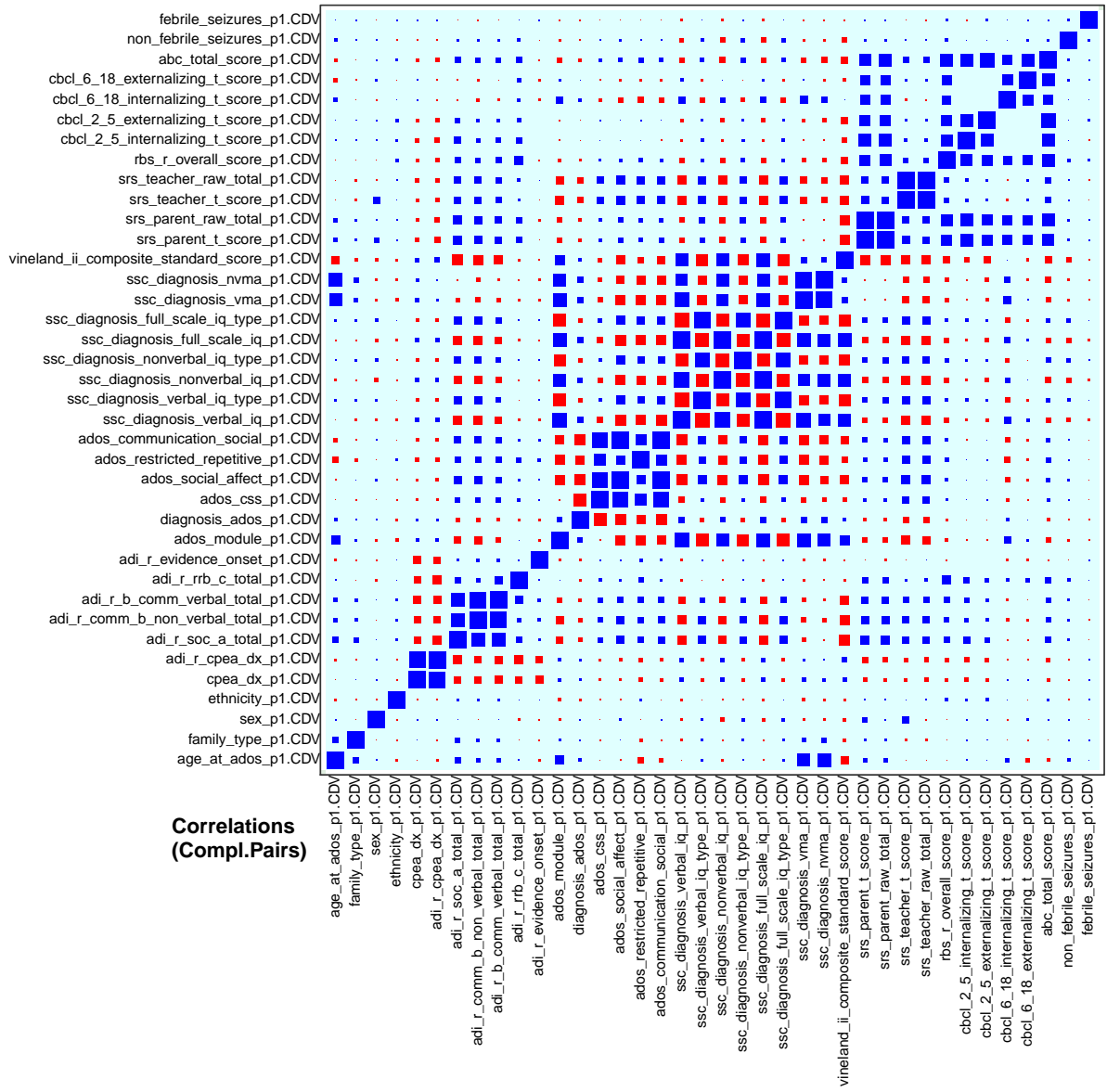


Figure 1: A first example of a “blockplot”: labels in the bottom and left margins show variable names, and blue and red blocks in the plotting area show positive and negative correlations.

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 precision and certainty of inference, whereas increasing p has the contrary effect of reducing the precision and certainty of inference. 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 rather crudely, such as with plain correlations. Correlations indicate the basic directionality in pairwise association, and as such they answer the crudest 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 the Appendix we refute or temper each of these objections by showing (1) that correlations are usually useful measures even when the associations are non-linear, (2) that higher-order associations play a relatively minor role especially in the large- p problem, 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 applied 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**, implements the essential displays and interactive functionality to support the “mining” of large correlation tables as proposed here. The software is written entirely in the **R** language¹, but it relies on a function that is currently implemented only for the *Microsoft Windows* operating system² (this should change in the not too distant future).

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 population dataset used in this document by applying at <https://base.sfari.org>.

¹<http://www.cran.r-project.org>

²*Microsoft Windows* is a trademark of Microsoft, Inc.

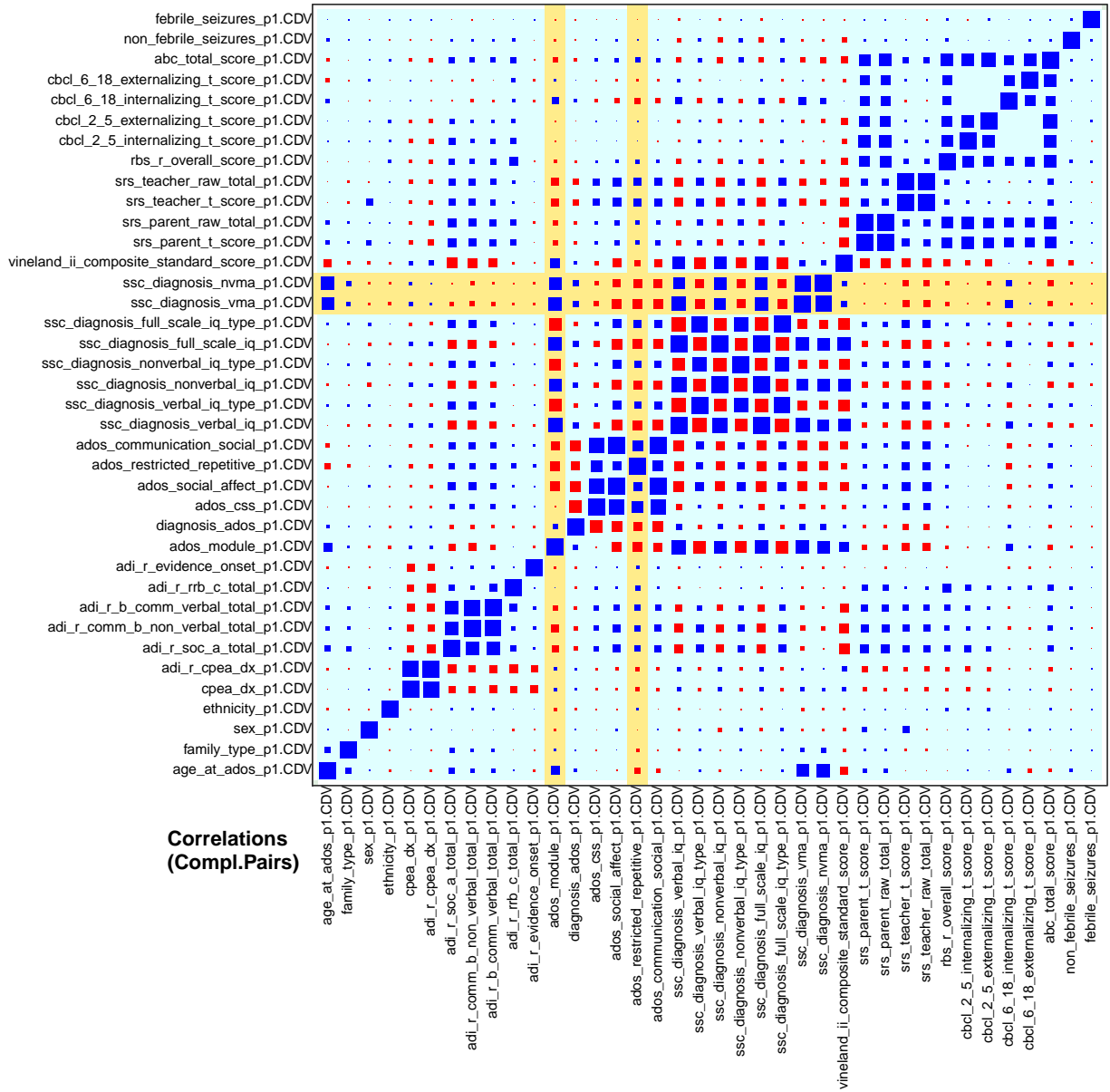


Figure 2: A “reading exercise” illustrated with the same example as in Figure 1: The salmon-colored strips highlight the variables `age_at_ados_p1.CDV` and `ados_restricted_repetitive_p1.CDV` on the horizontal axis, and the variables `ssc_diagnosis_vma_p1.DCV` and `ssc_diagnosis_nvma_p1.DCV` on the vertical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

2 Graphical Displays

2.1 Graphical Display of Correlation Tables: Blockplots

Figure 1 shows a first example of what we call a “blockplot”³ 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 (partly following the financial convention where “being in the red” implies negative numbers). — 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 when 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 tool (see Sections 2.3 and 2.4).

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 axis⁴:

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

and two other variables on the vertical axis:

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

(For readability we shorten the labels in what follows.) In the intersection of the left vertical strip with the horizontal strip, we find two blue blocks of which the lower block is recognizably larger than the upper, implying that the correlation of `age_at_ados..` is positively correlated with both `ssc_diagnosis_vma..` and `ssc_diagnosis_nvma..`, but more strongly

³ This type of plot is sometimes called “fluctuation diagram” (H. Hofmann, *Metrika* 2000, 51, 11-26). The term “blockplot” is ours, and we introduce it because we think it is more descriptive of the plot’s visual appearance.

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 variable”.

with the former than the latter. — 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 `ados_restricted..` is negatively correlated with both `ssc_diagnosis_vma..` and `ssc_diagnosis_nvma..`, but more strongly with the former, and both are more weakly correlated with `ados_restricted..` than with `age_at_ados...`

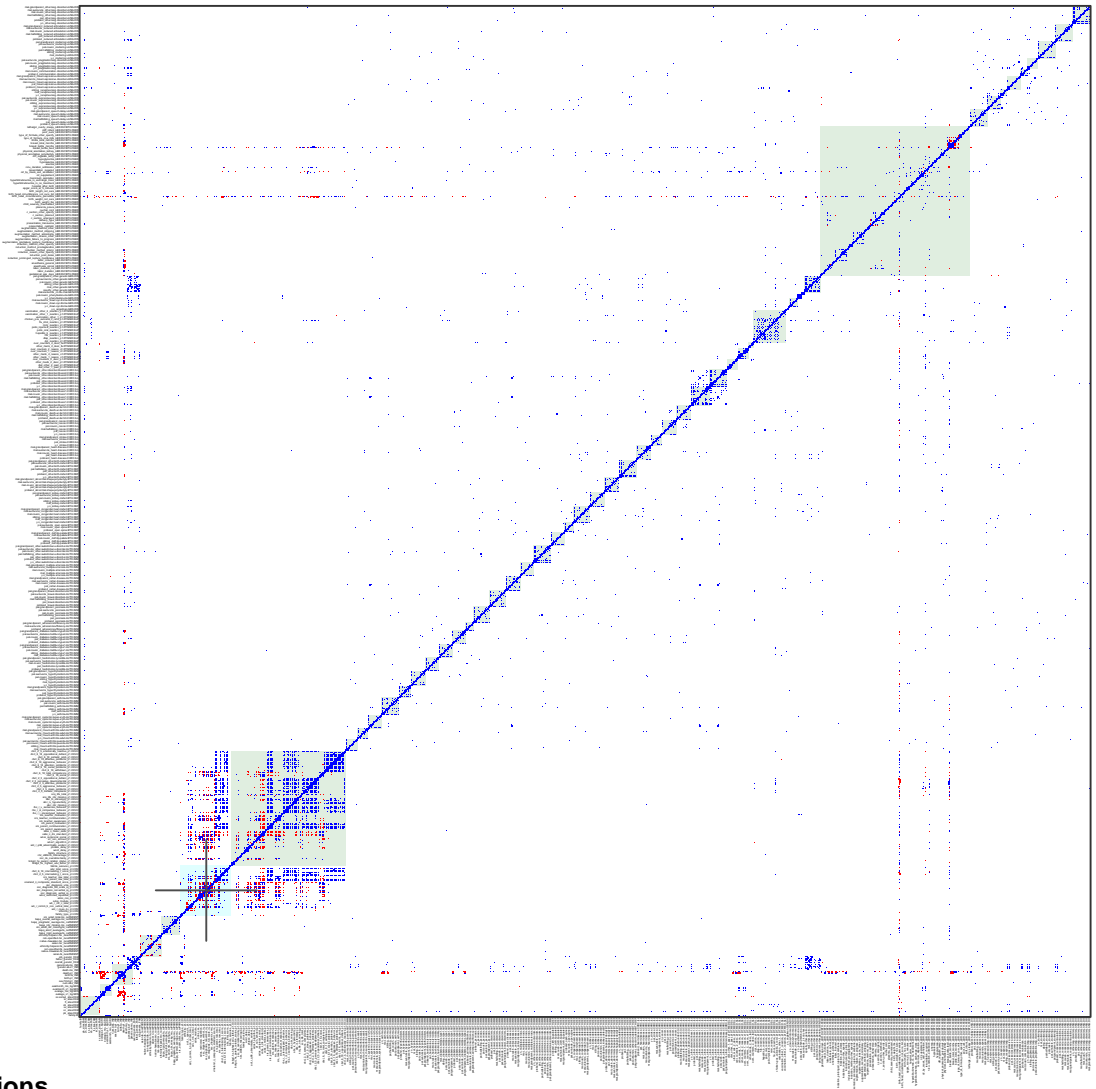
2.2 Graphical Overview of Large Correlation Tables

Figure 1 shows a manageably small set of 38 variables. This, 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.

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, pointed at by a faint crosshair and shown in turquoise as opposed to gray: this square marks the group of 38 variables shown in Figure 1.

The mechanism by which variable grouping is conveyed to the tool 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 “_”, 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.

⁵ These suffixes were chosen to 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 drawn from the following SSC tables: `proband_cdv.csv`, `proband_ocuv.csv`, and `parent.csv`.



**Correlations
(Compl.Pairs)**

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.*

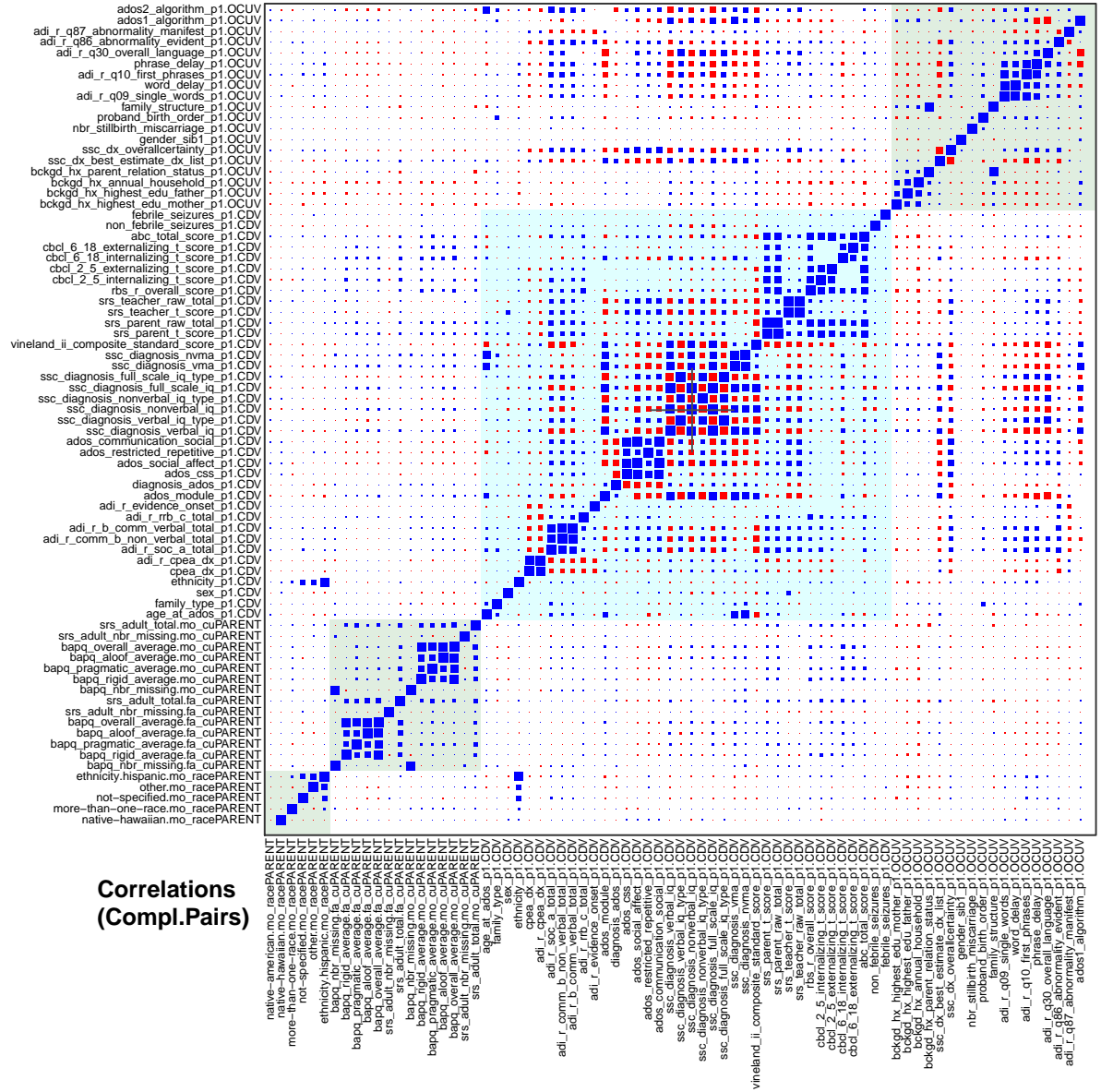


Figure 4: The 38 variable group of Figure 1 in the context of the neighboring variable groups.

2.3 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 zero⁶. 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 population correlations vanish. Such protection is provided by choosing a threshold much smaller than 0.05, by one conservative rule, named after “Bonferroni”, as small as $0.05/\#\text{correlations}$. 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-values⁷.

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 which 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 $1-(\text{p-value})$.
- Drawing on the above 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 variable 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

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.

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”.

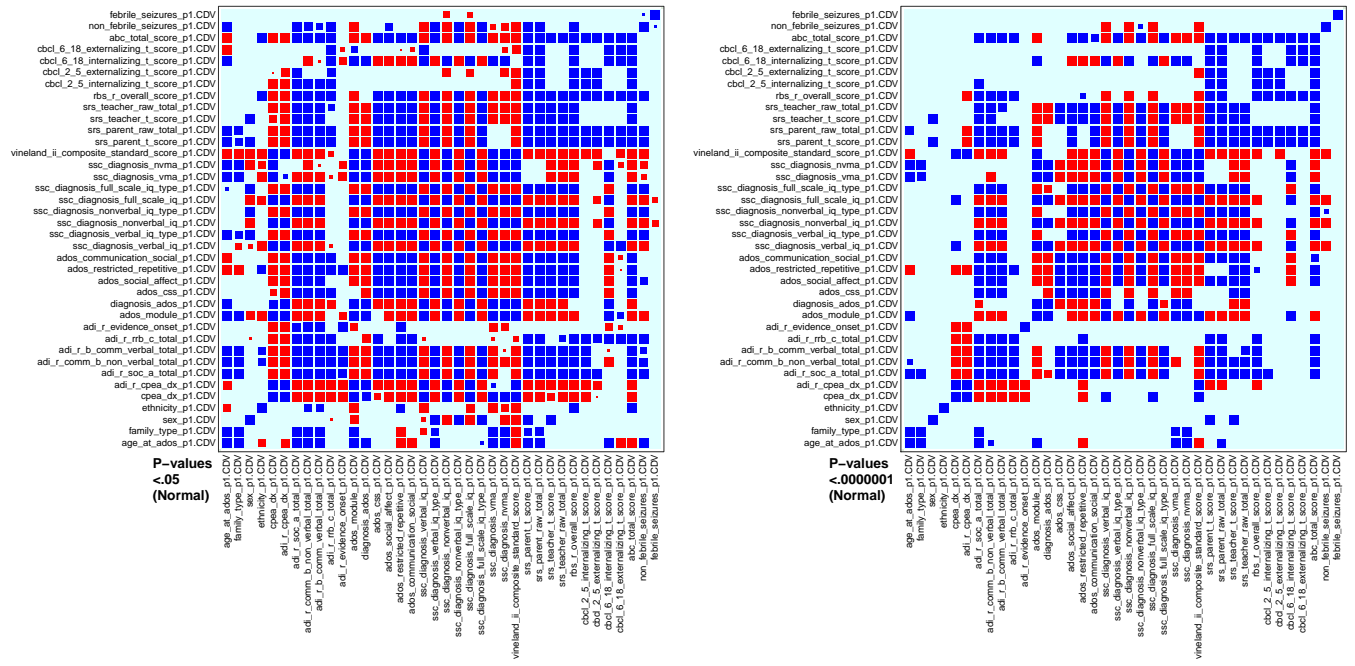


Figure 5: Blockplots of the p -values for the 38 variable group of Figure 1. Smaller and hence statistically more significant p -values are shown as larger blocks. The colors are inherited from the correlations to reflect their signs.

Truncation levels of p -values: Left ≥ 0.05 ; right $\geq 0.000, 000, 1$.

Many modest correlations are extremely statistically significant due to $n \geq 1, 800$.

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 it is going out on a limb to rely 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, even with the problems of multiplicity and extreme tail probabilities in mind.

2.4 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, 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.

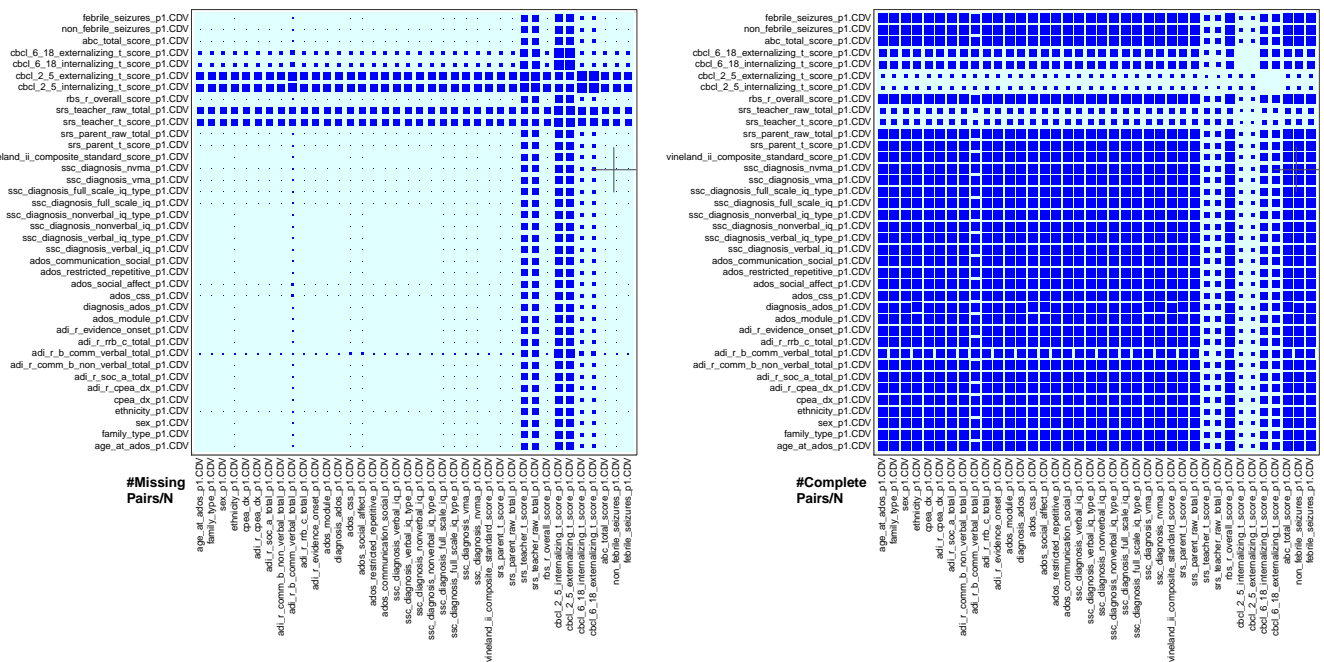


Figure 6: Blockplots of the fractions of missing (left) and complete (right) pairs of values.

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-a-missing values in both variables. Thus, associated with each correlation $r(x, y)$ are

- the number $n(x, y)$ 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.

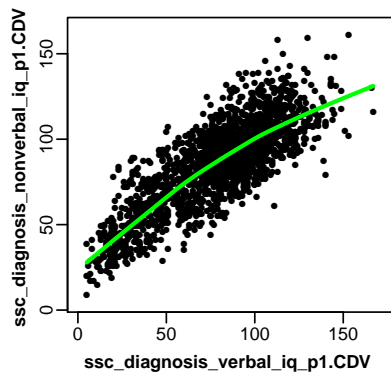
Depending on whether the number of complete or incomplete pairs dominates, one or the other plot is more sensible in that it uses less ink. Finally, we note that in this case of a blockplot the diagonal is not occupied by a constant but contains instead the fraction of non-missing ($n(x, x)/n$) and missing ($m(x, x)/n$) 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.

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

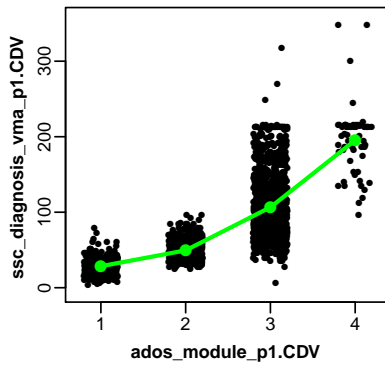
The correlation of a pair of variables is a crude measure of association between two variables, hence one often wonders about the detailed nature of the association. The full details can be learned from a scatterplot of the two variables. Often the association is constrained by the marginal distribution, 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 and the y -variable in the center are **quantitative**, the x -variable in the center is apparently **ordinal** with four values, and both variables on the right 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 bimodal⁸, with a major mode around $x = 90$ and a minor mode around $x = 30$. The y -variable in the center 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,

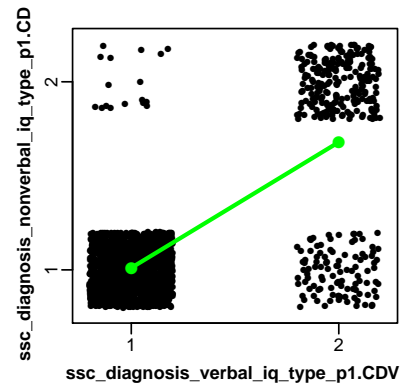
⁸ The bimodality might have a trivial explanation, which we wouldn't be able to rule out based on the given data: there might be a dominant binary variable lurking in the composition process that produces verbal IQ. If a lurking variable could be ruled out, the bimodality might be of substantive interest.



Frame #1: n = 1887
Corr = 0.824 (pval = 0)



Frame #2: n = 1880
Corr = 0.707 (pval = 0)



Frame #3: n = 1887
Corr = 0.76 (pval = 0)

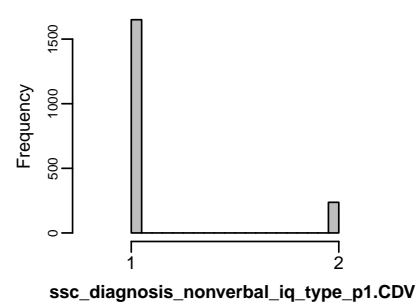
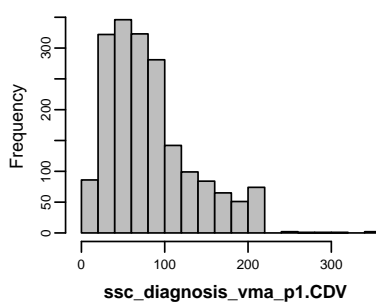
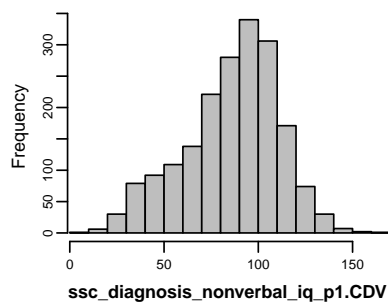
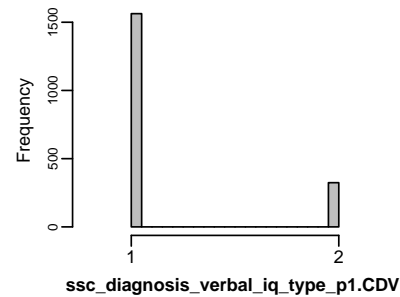
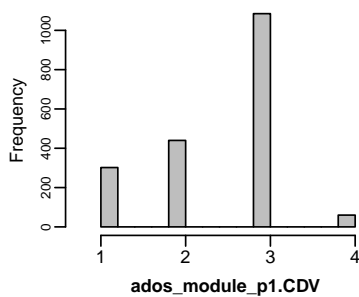
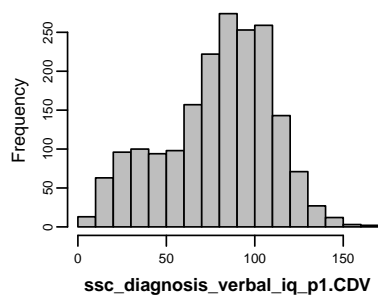


Figure 7: Scatterplots and histograms/barplots for three variable pairs.

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 enhanced with **smooths** for continuous variables and with **traces of group means** when the x -variable is categorical with fewer than, say, 8 groups. 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 modest⁹.

The tool 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 tool 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 familiar with scatterplot matrices, note that somewhat unusually 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.

2.6 Variations on Blockplots

Blockplots as used here are not the conventional visualizations of correlation tables. As a [google search of “correlation plot”](#) shows, the most frequent visual rendering of correlation tables is what we may call “checkerboard plots”, as opposed to “blockplots” shown so far. Checkerboard plots encode the magnitude of correlations in terms of color intensity of fixed-sized squares. An example is shown in the left frame of Figure 9; for comparison, the right frame shows a blockplot. Here are a few observations about the two types of plots:

- Color 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.
- In checkerboard plots color bleeds between adjacent squares and fuses them. 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 checkerboard plots.

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.

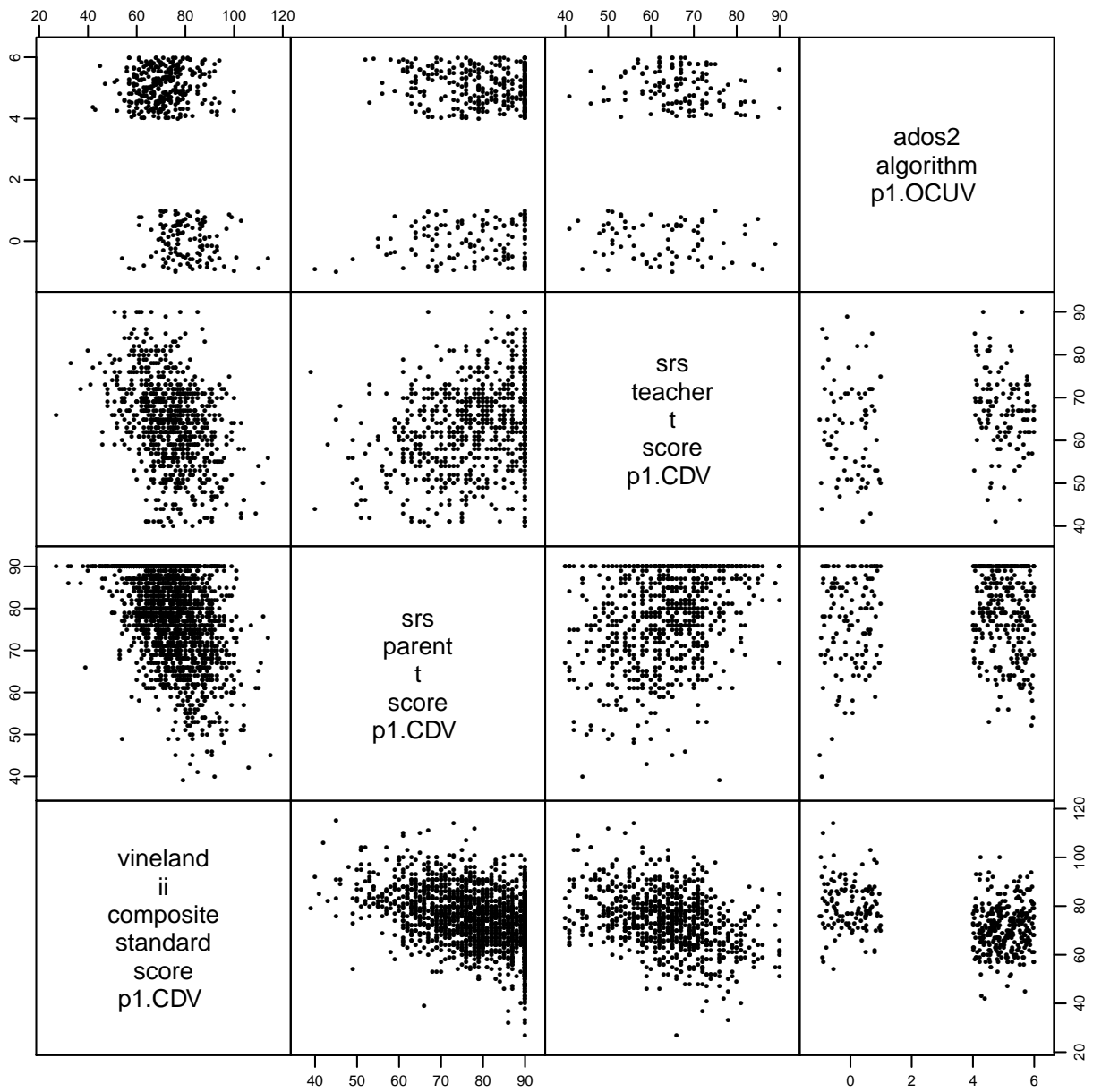


Figure 8: Scatterplot matrix of four variables. (Note the convention for the vertical order of the variables: bottom to top, for consistency with the blockplots.)

- Checkerboard plots do not permit mark-up with background color because they fill the plotting surface completely. This problem can be overcome by shrinking the checkerboard squares 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, gives up some of the crispness that the blockplot and to a lesser degree checkerboard plots have.
- The checkerboard plot or its shrunk version in the center frame of Figure 9 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. On raster screens the pixel resolution also sets a finite lower limit on blocks and resizing them.

Because none of the two types of plots — blockplots or checkerboard plots — are uniformly superior at all scales, the **AN** provides both, and with one keystroke one can toggle between the two rendering methods.

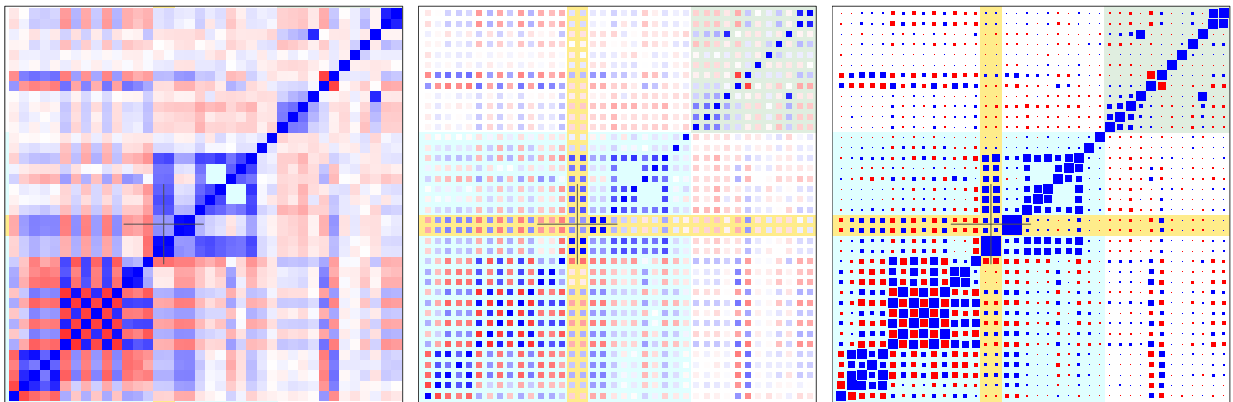


Figure 9: A “checkerboard plot” (left) compared with a corresponding blockplot (right), as well as a “shrunk checkerboard plot” in the center.

3 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. The 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, 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.

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

```
source("http://stat.wharton.upenn.edu/~buja/SFARI-875253/association-navigator.R")
mymatrix <- matrix(rnorm(20000),ncol=100)
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. The reader may left-drag the mouse in the plot to see a first realtime response.

3.1 One-time Setup

The version of the **AN** presented here is intended for the SFARI community. The tool comes therefore packaged with a massaged version of the SSC phenotype data (Version 9, August 2010), described in Appendix 2 below. The software and the data are available as a **R** workspace file that can be downloaded from the internet and loaded into **R** in one of the following ways¹⁰:

- If the reader is viewing the PDF file of this document, he/she may click on this address (or else type it into the address field of a browser):

<http://stat.wharton.upenn.edu/~buja/SFARI-875253/AN-SSC.RData>

It is recommended not to open but to save the workspace in a permanent folder of the user's choice. To start up **R** and load the workspace, the user may double-click the file in an explorer directed at the appropriate folder. (This takes a few seconds as the workspace expands into a few hundred megabytes when loaded.)

- An alternative is to start a **R** interpreter and type or copy/paste the following into it:

```
load(url("http://stat.wharton.upenn.edu/~buja/SFARI-875253/AN-SSC.RData"))
```

¹⁰ We assume that the **R** software is installed on the reader's machine: <http://www.r-project.org/>

(This also takes a few seconds to complete, and it will generate a warning about closing the URL, which can be ignored). Care should be taken to save the workspace by either typing

```
save.image()
```

into the **R** interpreter, or, when quitting **R**, answering the question “Save workspace image? [y/n/c]” with “y”. This saves a local workspace file `.RData` in the current folder (also taking a few seconds to complete).

This concludes the one-time operations necessary to load the **AN** and the SSC data. The following sections describe recurrent operations that apply to each **R** session with **AN** use.

3.2 Starting Up the AN

- **Starting R with the AN loaded:** The simplest method is to double-click the saved `.RData` or the downloaded `AN-SSC.RData` file in an explorer window¹¹.
- **Starting the AN:** Type in the **R** Gui

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

A new window will appear with a blockplot of the 757 variables shown in Figure 3 above. Loaded into the workspace is an instantiation (`a.n`) of the **AN** with some joined SSC tables as its data. The function that brings the **AN** to life is `a.nav.run()`.

- **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 letter ‘q’ into the **AN** master window. The master window will remain as a passive **R** plot window, but it will no longer respond to user input. On the other hand, 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.

¹¹Else the user can start **R** in any other way and load the workspace with the second square icon in the **R** Gui or with the `load()` function.

- **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. This holds even between **R** sessions if the previous session had been saved.

3.3 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.

- **Crosshair:** To place it, simply left-click 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. Then place the crosshair on one of the diagonal highlight squares. This will be the target for zooming, which is 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 approximately to the highlight square where you placed the crosshair.

- **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, 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.

3.4 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 sufficiently.

- Crosshair size: hit or depress
 - ‘c’ to decrease,
 - ‘C’ to increase.

Exploding the crosshair by depressing ‘C’ can be an effective alternative to highlight strips (Section 3.6) for visually connecting a block to its x - and y -variable labels.

- 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.

3.5 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-0’ for observed correlations,
- ‘ctrl-P’ for p-values of the correlations (Section 2.3),
- ‘ctrl-M’ for fraction of missing/incomplete pairs (Section 2.4),
- ‘ctrl-N’ for fraction of complete pairs (Section 2.4),.

As discussed in Section 2.3, 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.

3.6 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.

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 3.3).

3.7 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 removed.

After ‘5’ the user can reshape the whole blockplot window to approximately fit the aspect ratio of the highlight rectangle.

- 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.

3.8 Reference variables

A recurrent problem when using the **AN** is that some variables are of persistent interest. In autism phenotype data, for example, a recurrent theme is to check up on age, gender and site association 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 3.6), 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.

In Figure 10, for example, 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_adol_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.

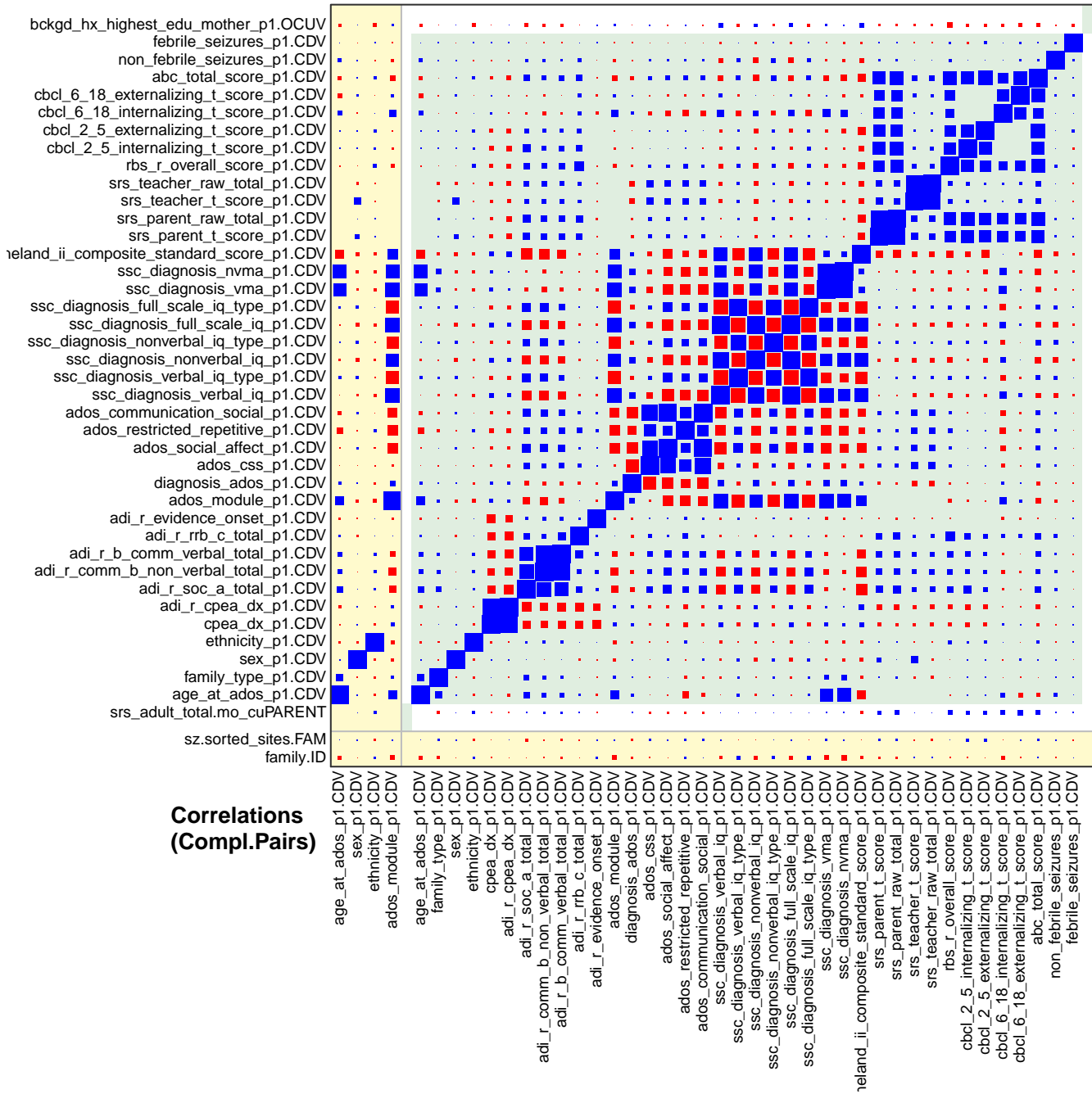


Figure 10: Reference variables shown in the left and bottom bands. Whenever the user zooms and pans the blockplot, these variables stay in place and show their associations with the variables from the rest of the blockplot.

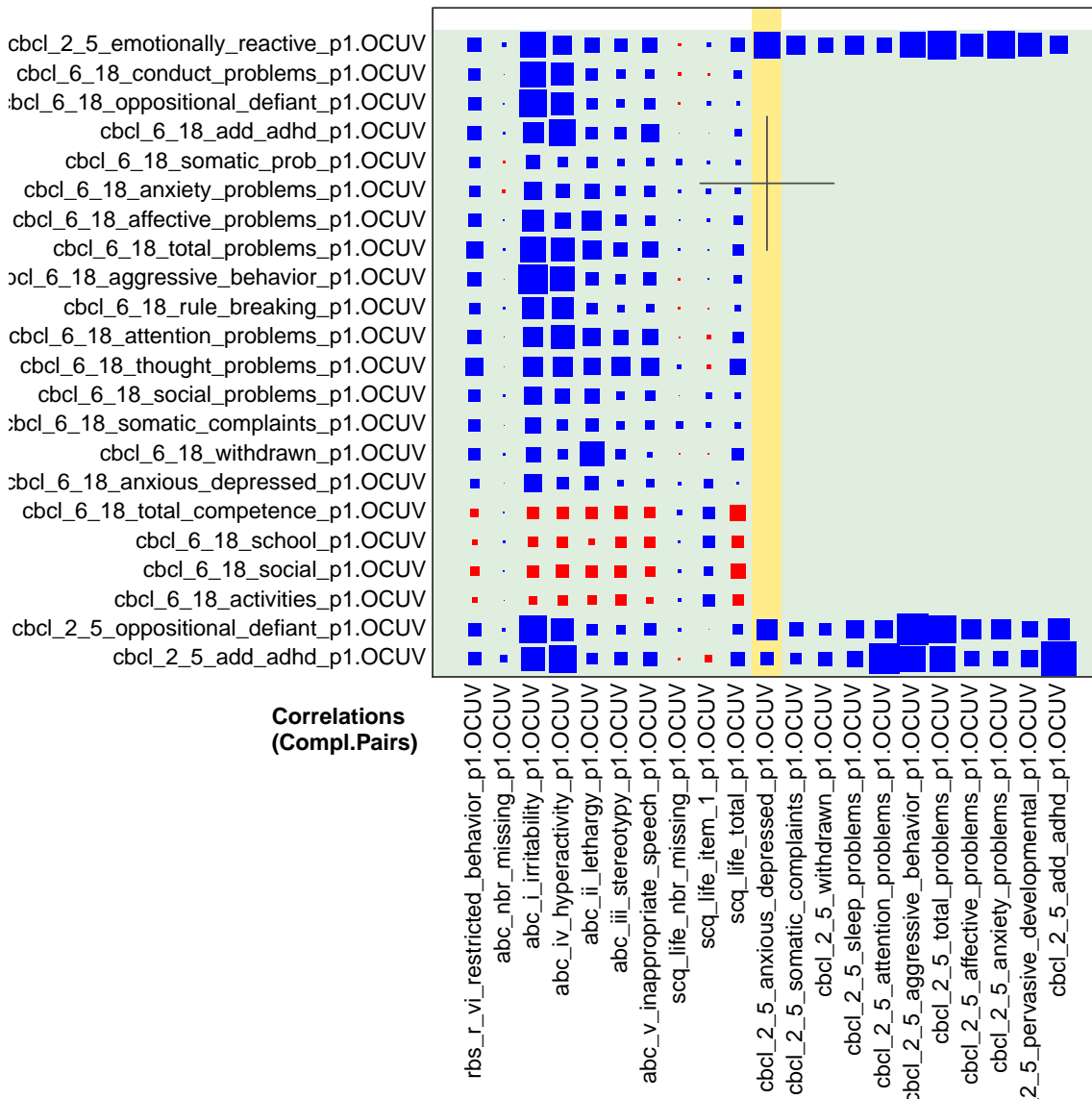
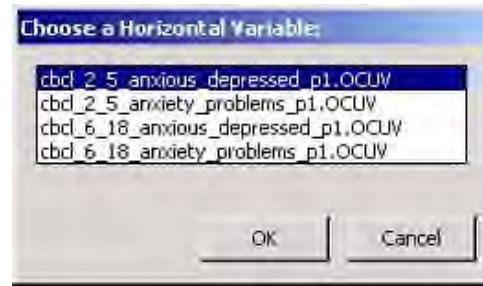
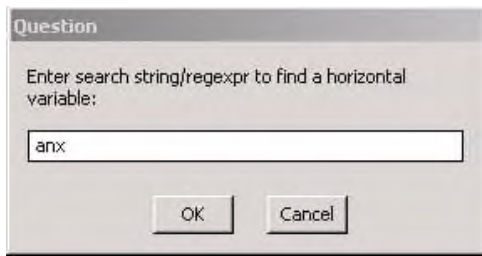


Figure 11: Text search with ‘H’ for horizontal variables containing “anx”, followed by selection of “cbcl_2.5_anxious.depressed_p1.OCUV”. The view pans horizontally to the selected variable, marks it with a vertical highlight strip, and places the crosshair on it.

3.9 Searching Variables

Other recurrent problems 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.

3.10 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 2.5 above. The mechanics are as follows: Hit

- ‘**x**’ to see in a separate window (Figure 7) a scatterplot and barplots/histograms of the two variables marked by the crosshair cursor.
Important: The lense window is passive and does not accept interactive input. One must expose the blockplot master window to continue with **AN** interactions.
- ‘**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.

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 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.

3.11 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.
- 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.

3.12 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

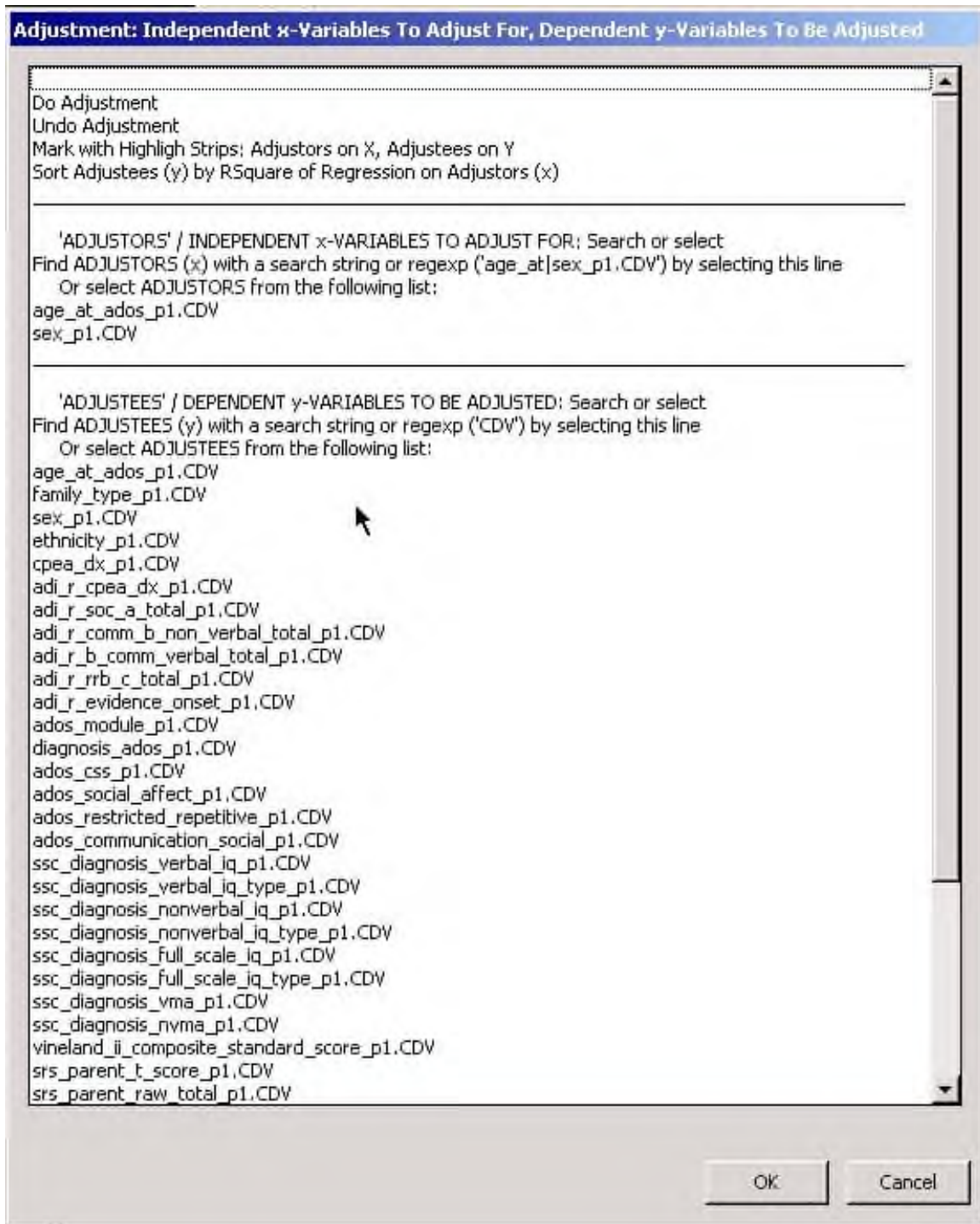


Figure 12: Screenshot of the adjustment menu. As shown, it enables adjustment of the “srs” variables for “age_at_ados_p1.CDV” and “sex_p1.CDV”.

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 autism phenotype, for example, one may be interested in adjusting for age and/or gender. In practice, particularly in large- p problems, there is rarely sufficient data to truly estimate conditional distributions¹², hence one makes the simplifying assumption that all associations are linear with constant conditional variances (homoscedasticity)¹³. 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: $y_{\bullet x} = y - (b_0 + b_1x)$. 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 but menu selection, similar to Section 3.9, 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 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 3.8) 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.

¹²Natural exceptions do exist: If we analyze females and males separate, for example, we study gender-conditional associations.

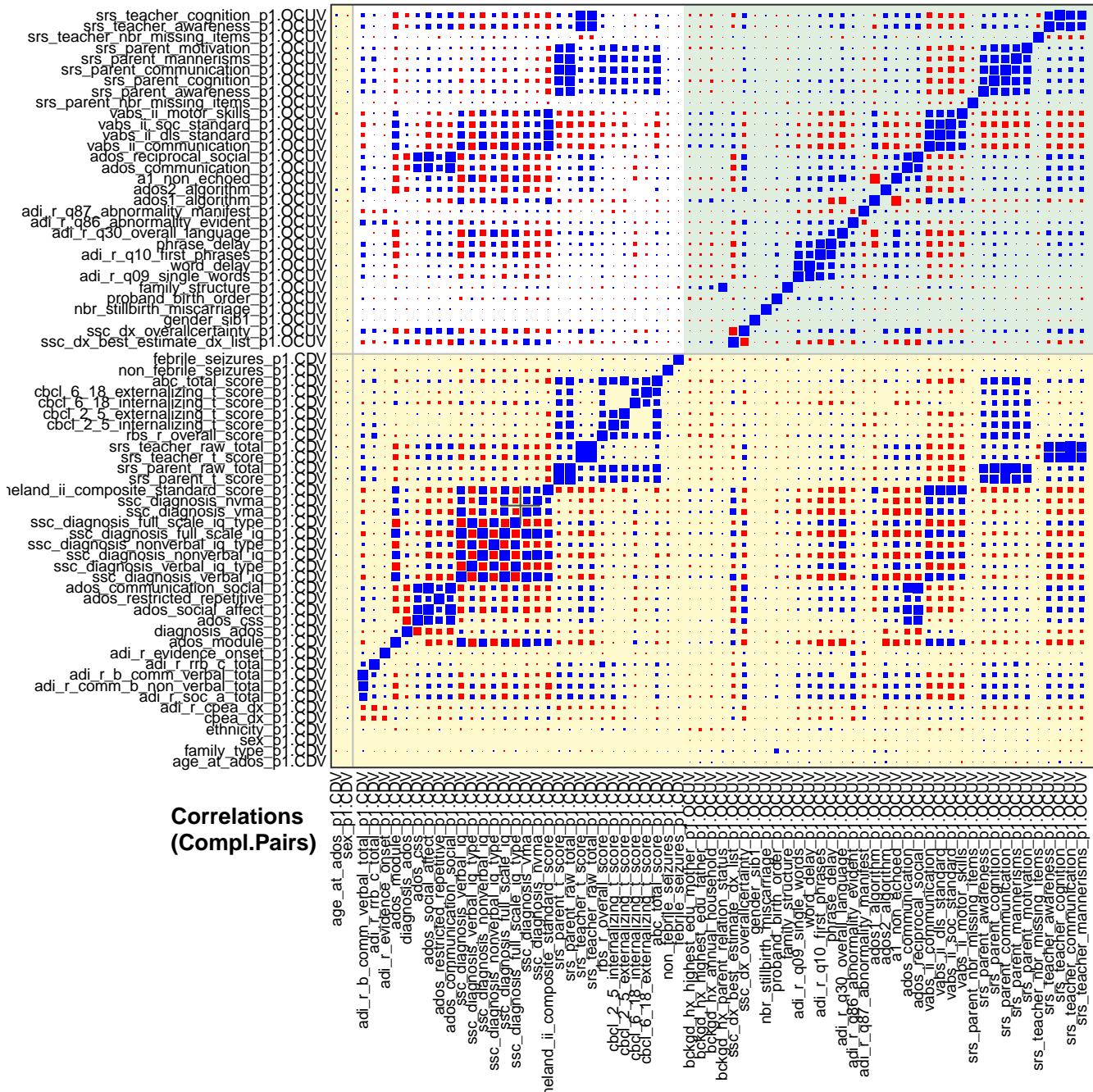
¹³Both assumptions may be wrong, but some form of adjustment, even if flawed, is often more informative than remaining with raw variables.

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 3.5) or by looking at scatterplots (Section 3.10).

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.

3.13 Missing Functionality

The functionality described here has been implemented in the September 2010 version of the AN. Some 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.



**Correlations
(Compl.Pairs)**

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.

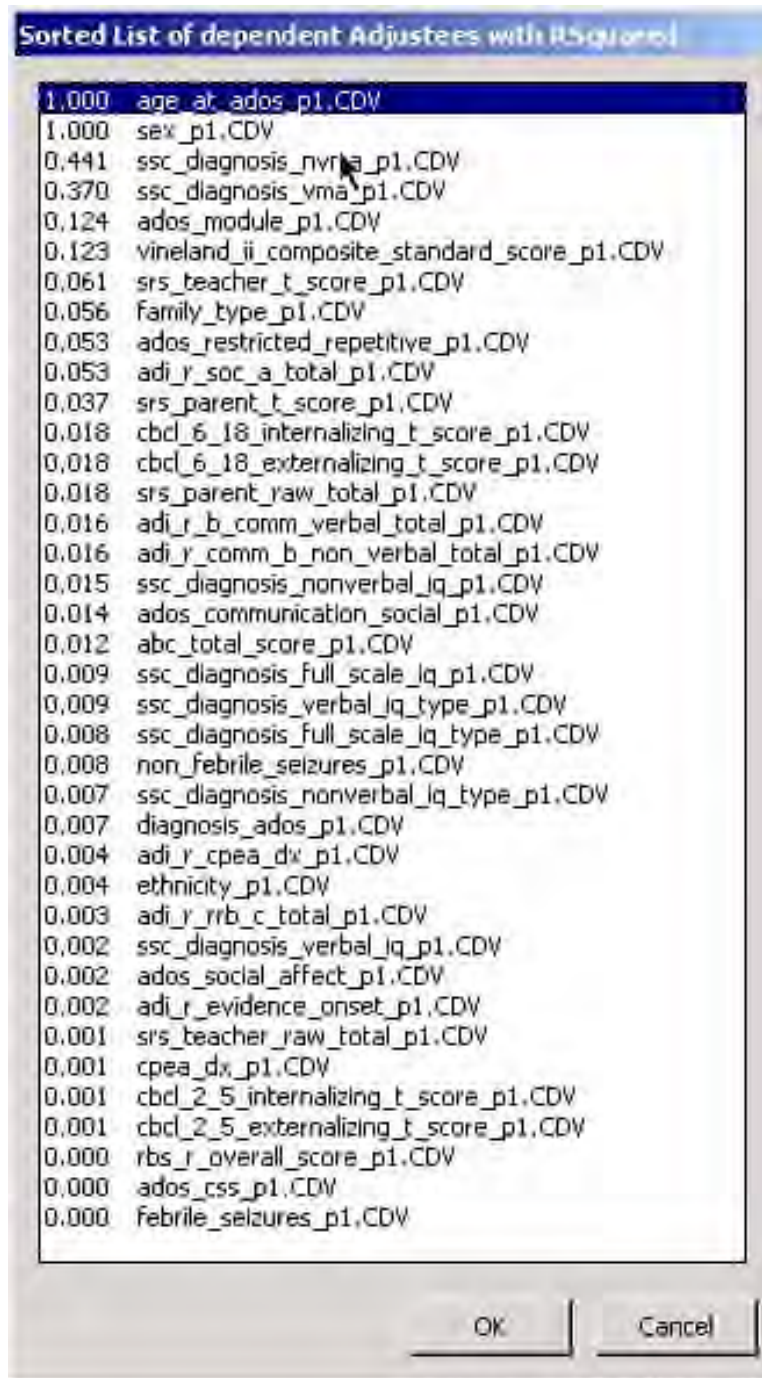


Figure 14: List of adjustees/*y*-variables sorted according to the R^2 values from the regressions onto the adjustors/*x*-variables.

Appendix 1: 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 linear¹⁴
- (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 variables¹⁵. 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 contexts¹⁶.

¹⁴ 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, $E[Y|X] = \beta_0 + \beta_1 X$, 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-examples 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 .

¹⁵ An example would be three variables jointly uniformly distributed on the surface of a 2-sphere in 3-space.

¹⁶ 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 procedures¹⁷. 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.

¹⁷ For example: De Leeuw, J. and J. van Rijkevorsel (1980). “HOMALS and PRINCALS - Some generalizations of principal components analysis,” p. 231-242, in E. Diday et al., editors. *Data Analysis and Informatics II*. Elsevier Science Publisher, North Holland, Amsterdam.
 Gifi, A. (1990). *Nonlinear multivariate analysis*. Wiley: Chichester.

- 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.

Appendix 2: 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 **ANs** that display large numbers of SSC phenotype variables. The reader is warned that as written the statement would produce an **AN** in the order of 3,000 variables. It is therefore recommended to comment out tables that are not of immediate interest and to form an **AN** from a more manageable subset of tables. Most of the tables will be recognizable to the autism phenotype expert. In most cases they are slightly massaged versions of corresponding SSC tables with similar names. The massaging consists mostly of eliminations of administrative variables and the conversion of categorical variables to numeric ones either through scoring of ordinal variables or dummy coding of nominal variables (Appendix 1). The very first variable, “`family.ID`”, is a numeric coding of the administrative family ID used in the SSC; it is a potential indicator of lurking variables as it may relate to time order and site.

For the interested reader, **ANs** are implemented not as lists but as **R** environments. As such, one can look inside an **AN** with the **R** idiom

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

```

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.psu.h.fa, vv.ssc.psu.h.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.*

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

```
with(a.n, strips.col <- "mistyrose")
```