AgEcon SEARCH
RESEARCH IN AGRICULTURAL & APPLIED ECONOMICS

The World's Largest Open Access Agricultural & Applied Economics Digital Library

This document is discoverable and free to researchers across the globe due to the work of AgEcon Search.

Help ensure our sustainability.

Give to AgEcon Search

AgEcon Search
http://ageconsearch.umn.edu
aesearch@umn.edu

*Papers downloaded from **AgEcon Search** may be used for non-commercial purposes and personal study only. No other use, including posting to another Internet site, is permitted without permission from the copyright owner (not AgEcon Search), or as allowed under the provisions of Fair Use, U.S. Copyright Act, Title 17 U.S.C.*

No endorsement of AgEcon Search or its fundraising activities by the author(s) of the following work or their employer(s) is intended or implied.

heck for updates

The Stata Journal (2024)
24, Number 2, pp. 329–361

DOI: 10.1177/1536867X241258010

Speaking Stata: The joy of sets: Graphical alternatives to Euler and Venn diagrams

Nicholas J. Cox
Department of Geography
Durham University
Durham, U.K.
n.j.cox@durham.ac.uk

Tim P. Morris
MRC Clinical Trials Unit
University College London
London, U.K.
tim.morris@ucl.ac.uk

Abstract. Membership of overlapping or intersecting sets may be recorded in a bundle of $(0, 1)$ indicator variables. Annotated Euler or Venn diagrams may be used to show graphically the frequencies of subsets so defined, but beyond just a few sets such diagrams can be hard to draw and use effectively. This column presents two new commands for graphical alternatives: `upsetplot` and `vennbar`. Each command produces a bar chart by default, but there is scope to recast to different graphical forms. The differences between the new commands reflect the divide in Stata between `twoway` commands and other `graph` commands. They also provide some flexibility in graph design to match tastes and circumstances. The discussion includes many historical details and references.

Keywords: gr0095, upsetplot, vennbar, bar charts, Venn diagrams, Euler diagrams, UpSetPlots, set membership, indicator variables, binary variables, graphics

1 The problem: Indicating set membership graphically

The problem surveyed in this column is elementary, which as in logic or physics means fundamental as much as simple. In many projects, we have overlapping or intersecting sets. Suppose that data on membership of each set are held in a bundle of indicator variables, each with value 1 if an observation belongs to a particular set and value 0 otherwise. We want to show the frequencies of different subsets graphically.

Typically, counting to get the frequencies is easy. Often, the frequencies are already in a separate variable in a dataset, which solves that problem immediately. The problem lies in the graphics; we want to produce a display that is not only easy to understand in principle but also effective in conveying both coarse and fine structures in practice.

Concrete examples arise in many fields.

Missing values. Many projects need to confront serious missingness. We can represent missingness by indicator variables: 1 for missing and 0 otherwise. This can be done for both numeric and string variables. We start with examining frequencies of missing values in each variable and continue with frequencies of joint occurrences, examining how far missing data occur in blocks and so forth.

© 2024 StataCorp LLC

Medical symptoms. Many diseases, syndromes, or other medical conditions manifest in different ways. Patients diagnosed as having a particular condition do not necessarily manifest the same symptoms.

Social survey. Indeed, people can often be recorded using a battery of categorical variables. Many categorical variables are binary (employed or not employed, living in city or not, college graduate or not). Here we note just once that variables with multiple categories (meaning three or more categories) can be recorded in terms of a bunch of indicator variables, one fewer in number.

Gene families in genomes. Overlap between gene families in various genomes deserves mention here for its own sake and because it is a field of study that has sparked innovation over graphics for set membership.

Network meta-analysis. Network meta-analysis compares three or more interventions simultaneously by synthesizing the results of studies that have compared at least two. Understanding and visualizing the structure of a network meta-analysis can be challenging. The numbers of studies or of people contributing to direct evidence on a given set of interventions are important summaries. Freeman et al. (2023) proposed some visualizations, including upsetplots, for *component* network meta-analysis.

This column is focused on two new commands: `upsetplot` and `vennbar`. Why two commands are offered rather than one hinges partly on the structure of `graph` commands in Stata. It also serves to offer some variety of graph style. Section 2 discusses the problem generally but in more depth and detail. Sections 3 and 4 explain and exemplify the use of `upsetplot` and `vennbar`, respectively, and sections 5 and 6 give more formal statements of their syntax. Note that the help files of both commands give many more examples than are discussed here. Section 7 contains various historical and bibliographical remarks.

Talking about two commands that are similar, but certainly not identical, obliges a modest amount of repetition. No reader will find all sections equally interesting or useful, but every reader will be smart enough to skim and skip according to inclination.

Combined tabulations of multiple indicator variables are possible through the `groups` command. See Cox (2017, 2018) and `search st0496, entry` to find any later update.

2 General principles and precepts

2.1 Euler–Venn diagrams and the combinatorial challenge

A common solution for a few sets is to annotate an Euler–Venn diagram. Later, we will give more historical and bibliographical details, but for now a brief comment: Euler is named here as well as Venn as a small pointer to a longer history than is implied by the most common name, *Venn diagram*.

Whatever the name, we assume readers are familiar with such diagrams. If not, please take a few moments to Google for some explanations and examples. There is no official Stata command for Euler–Venn diagrams, but community-contributed commands include those from Lauritsen (1999a,b,c,d, 2009), Gong and Ostermann (2011), and Over (2022).

Most crucially, it is common experience that, while such diagrams can be helpful for very simple problems, they become unwieldy for more than a few sets. This is perhaps too obvious to deserve emphasis, but here are some quotations to that effect.

Hamming (1991, 16–17) commended Euler–Venn diagrams for simple cases yet continued as follows:

> *But if you try to go to very many subsets then the problem of drawing a diagram which will show clearly what you are doing is often difficult. Circles are not, of course, necessary but when you are forced to draw very snake-like regions then the diagram is of little help in visualizing the situation.*

Gleason (1991, 33) noted that, in practice, the diagrams become unwieldy for more than about four or five sets.

Kosara (2007) was particularly brutal:

> *I would argue that Venn diagrams are a great tool for learning about sets, but useless as a visualization.*

The point is better made by example than by exhortation. See figure 4 of D'Hont et al. (2012). We admire its wit but doubt its effectiveness. In principle, all the data on frequencies are shown. In practice, only individual detail can be read off at all easily. The comparison is of six genomes. Gene families that appear in none of the genomes in the study are not shown, so the total number of subsets is $2^6 - 1 = 63$, supporting Gleason's point.

The challenge here arises from elementary combinatorics. Given k sets and their indicator variables, there are 2^k possible subsets. Thus, for $k = 1, 2, \ldots, 5, \ldots, 10$, there are $2, 4, \ldots, 32, \ldots, 1024$ possible subsets. The number of possible subsets explodes as the number of sets increases.

In practice, this problem, sometimes called "combinatorial explosion", is eased whenever possible subsets do not occur and eased mightily when that happens often. Other way round, such explosive behavior underlines the importance of being able to select which subsets to show. The price of complexity may be to leave out rare subsets from a graphical display.

To keep track of the possible, or at least actual, subsets, users can code them by binary numbers (0 denoting absence and 1 denoting presence), such as 00, 01, 10, and 11 for $k = 2$. The concatenations 00, 01, 10, and 11 define the four possible subsets

defined by two variables, distinct binary codes for binary numbers 00 to 11, and distinct decimal equivalents 0 to 3.

Hence, concatenation is here a simple and natural way to define composite categorical variables (Cox 2007). 00 is of degree 0, 01 and 10 are of degree 1, and 11 is of degree 2. Here, and indeed generally, leading zeros are retained as helpful reminders even though they might be considered redundant or ornamental.

Similarly, three such variables have eight possible binary concatenations (000, 001, 010, 011, 100, 101, 110, and 111) and decimal equivalents (0 to 7). As remarked, k such variables define 2^k possible subsets.

The subset that is binary number zero (for example, 00 for $k = 2$) may or may not occur in the data. Sometimes it does, as with any patients with no symptoms or any people with no missing data, and sometimes it does not, as with many gene families that occur in none of the genomes in a study.

2.2 The great divide and its compensations

As often with Stata graphics, there is a choice here for programmers between writing a wrapper for, say, `twoway bar` on the one hand and `graph bar`, `graph hbar` or `graph dot` on the other. This choice is one reason for the provision of two separate commands, `upsetplot` and `vennbar`. We think of the choice between `twoway` commands and the other `graph` commands as the "great divide" in Stata graphics.

`upsetplot` is by default a wrapper for `twoway bar` and its siblings. There is scope to recast it in other forms: the most appealing are possibly to use `twoway spike` or `twoway dropline` instead of vertical bars.

`vennbar` is by default a wrapper for `graph hbar`. Similarly, it could be recast by (for example) calling up `graph bar` or `graph dot` instead.

The existence of two commands is a complication that we suggest is more positive than negative, affording greater choice of graphical style to meet as far as possible both researchers' tastes and what works best for particular datasets. In each case, options provide a great deal of flexibility. Care has been taken to provide similar syntax whenever the commands perform similarly. That eases switching between the two if researchers are unsure which is more suited to their current project, data, and audience.

2.3 Variables, values, and observations

`upsetplot` and `vennbar` both require a bundle of numeric variables with values 0 or 1. Such variables are variously called indicator, dummy, binary, dichotomous, zero-one, one-hot, Boolean, logical, or quantal. Missing values will be ignored. Presenting values other than 0 or 1 is considered an error. Observations used will thus have values 0 or 1 in all variables specified. Differently put, neither command is designed for string variables or categorical variables that have three or more distinct values.

If your dataset is already aggregated to frequencies or other measures of abundance, specify those as weights multiplying the indicator variables. Commonly, but not necessarily, integer frequencies define frequency weights, but both commands support measures with fractional parts, such as proportions or percentages, which can be specified as analytic weights.

The order of variables presented does not determine the order in which they are shown in a plot. By default, bars (spikes, dotted lines) are shown in order of subset frequency, but choosing a more suitable order is the user's prerogative. There is considerable scope to change that sort order using other criteria.

Variables that are identically 0 or identically 1, at least in the data being shown, are not always useful and so might be omitted. `findname` (Cox [2010], `search findname, sj` for updates) can be used to find such variables through option `all(@ == 0)` or `all(@ == 1)`. Such calls can be extended to check for missing values, which `upsetplot` and `vennbar` will ignore anyway.

Various `egen` functions can be useful in selecting observations of particular interest. Thus, `rowtotal()` yielding totals of two or more would identify occurrence of two or more conditions simultaneously.

2.4 Working with a reduced dataset

Like many graphics commands, these commands do various calculations first and use a reduced dataset in plotting the results of those calculations. In contrast, consider a scatterplot, where the user's point of view is that the quantities to be plotted are already variables in the dataset. The present problem is more like construction of a histogram, where the user's point of view is that—either by default or using explicit choices—the command should first calculate frequencies, fractions, percentages, or densities according to a set of bins with specified limits and then plot the histogram.

Unusually, the reduced dataset for both `upsetplot` and `vennbar` has variable names that are accessible to the user (and that are used in any saved version of the dataset). These variables fall into two distinct groups.

Variables for each subset in reduced data

`_binary` is string and contains a code such as `"00"`, `"01"`, `"10"`, or `"11"`.

`_decimal` is numeric and contains a decimal equivalent such as 0, 1, 2, or 3.

`_text` is string and contains a description using variable names or labels.

`_freq` is numeric and contains the frequency of occurrence.

`_percent` (optional) is numeric and contains percent occurrence.

`_degree` is numeric and indicates the number of participating sets.

Allenby and Slomson (2011, 14) comment: "There is, unfortunately, no standard notation for the number of elements in a set." They could have added "and no stan-

dard term either." Other terms encountered (other than "number of elements") include "cardinality", "order", "potency", "power", and the homely "size".

Variables for each set in reduced data

`_set` is string and indicates each set using its variable name or its variable label.

`_setfreq` is numeric and indicates the frequency of each set.

`_set` and `_setfreq` are physically but not logically aligned with the other variables in the reduced dataset.

3 upsetplot

Names should not matter, but they often do. A good name can be evocative, encouraging, or even entertaining. A poor name can be confusing or even condemn a good idea to obscurity.

The name *upsetplot* (or mutations using some upper case, an extra space, or both) was a play on *set*. The original author, Alexander Lex, was "upset" by certain Venn diagrams (Lex 2022). The term and the idea seem to have caught on widely within genomics, so we follow suit here.

Whatever you think of the name, note that the main idea was independently published by Unwin (2015, 179, 180, 182); see also Unwin (2024).

Our Stata implementation does not claim to provide or support all the extra bells and whistles implemented elsewhere, some of which seem likely only to complicate an already challenging design. Rather, it implements the core idea of a matrix legend to denote set membership and a summary of abundance for each set. That said, in some detailed respects, our implementation may allow better plots than some others.

3.1 Genomics example

The first example uses data from the study by Schnable et al. (2009) using counts of overlapping gene families for rice, maize, sorghum, and *Arabidopsis*. The frequencies here are copied from their figure 2, so, as often occurs, the counting is already done. We show how to enter the data directly as a set of indicator variables with their associated frequencies. *Arabidopsis* is a formal genus name, so we can use italic on our plots, following standard taxonomic practice.

```
. input Rice Maize Sorghum Arabidopsis freq
         Rice      Maize    Sorghum  Arabido_s       freq
  1.     1 0 0 0 1110
  2.     1 1 0 0 229
  3.     0 1 0 0 465
  4.     1 0 1 0 661
  5.     1 1 1 0 2077
  6.     0 1 1 0 405
  7.     0 0 1 0 265
  8.     1 0 1 1 304
  9.     1 1 1 1 8494
 10.     0 1 1 1 112
 11.     0 0 1 1 34
 12.     1 0 0 1 81
 13.     1 1 0 1 96
 14.     0 1 0 1 11
 15.     0 0 0 1 1058
 16. end
. label variable Arabidopsis "{it:Arabidopsis}"
```

We will use the same top title for the next few graphs. It is convenient to put the option text in a local macro. That is just general Stata technique and not at all a requirement of `upsetplot`.

The *Stata Journal* uses the `stsj` scheme and does not print in full color. In contrast, `upsetplot` by default uses a more colorful presentation. That is also true of `vennbar`. For this article, we first set some extra options to use marker color `gs8`.

```
. local toptitle "t1title(Number of gene families)"
. local paperopts matrixopts(mc(gs8 ..))
. upsetplot Arabidopsis Rice Maize Sorghum [fw=freq], varlabels
> baropts(`toptitle') `paperopts'
```

(*output omitted*)

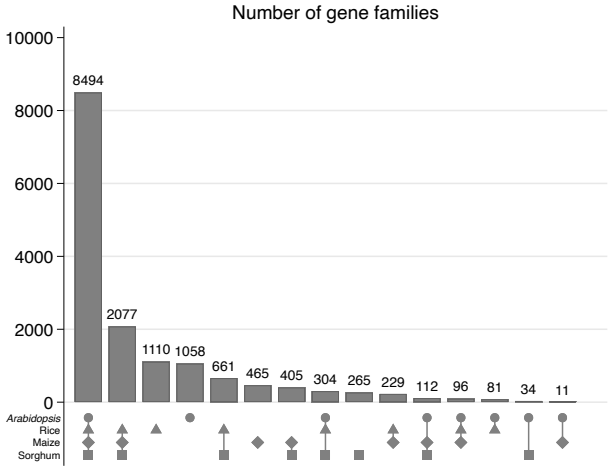

Figure 1. Default upsetplot for overlapping gene families. By default, subsets are ordered left to right by frequency.

Figure 1 is our first upsetplot. The default ordering of bars is left to right by frequency from most frequent to least frequent. The main twist on standard bar chart designs is use of a graphical legend in which a marker that is present denotes membership of a set.

As is common in statistical graphics, even with this relatively simple dataset, there is a little tension between a desire to show detail and a need to avoid crowding. The `varlabels` option allows the display of biologically correct italic, using the variable label just defined. Note that the option calls up variable names if no variable labels are defined.

Flexible control over sorting is strongly emphasized as a feature of `upsetplot`. Figure 2 shows the `gsort()` option in action. Its argument for that plot, `_decimal`, is a variable in the dataset by the time the graph is drawn. The name `gsort()` and the way it works are modeled on the `gsort` command, which is used inside the code. The point of `gsort`, as compared with the more often used `sort` command, is that the syntax allows minus signs, which reverse the order of sorting. In Stata, sort order for numeric variables is by default ascending order, lowest first. A minus sign in `gsort` syntax spells out that descending order is wanted. The order of the variables on the plot implies the order first absent on *Arabidopsis*, second present on *Arabidopsis*; within each of those two subsets, first absent on rice, second present on rice; and so on.

```
. upsetplot Arabidopsis Rice Maize Sorghum [fw=freq], varlabels
> baropts(`toptitle') `paperopts' gsort(_decimal)
```

(*output omitted*)

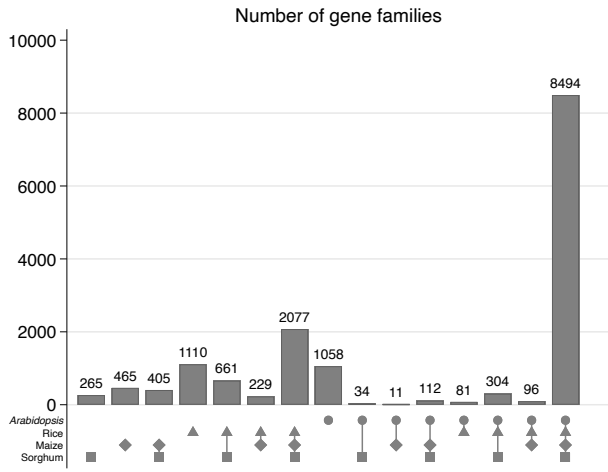

Figure 2. `upsetplot` for overlapping gene families. Sorting now respects the ordering of variable names.

```
. upsetplot Arabidopsis Rice Maize Sorghum [fw=freq], varlabels
> gsort(_degree -_freq) `paperopts' baropts(`toptitle')
```

(*output omitted*)

Figure 3 below sorts first on the number of sets to which a gene family belongs and then within that order by declining frequency.

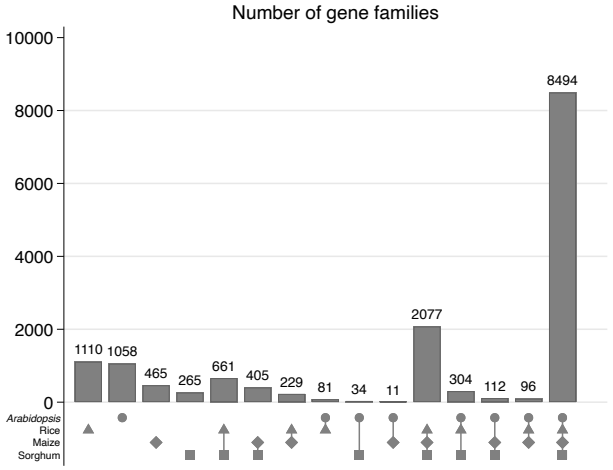

Figure 3. `upsetplot` for overlapping gene families. Sorting now respects ordering by number of sets belonged to in descending order and then frequency in descending order.

Although the result is not shown here to save space, the next command shows how to get a twist on that, reversing the order on the first criterion.

```
. upsetplot Arabidopsis Rice Maize Sorghum [fw=freq], varlabels
> gsort(-_degree -_freq) `paperopts' baropts(`toptitle')
```

(*output omitted*)

If you are following along on your computer, it would be a good idea now to

```
. save arms
file arms.dta saved
```

As explained in section 2.4, `upsetplot` computes and displays variables `_set` and `_setfreq`, which summarize the input data at set level. With the `savedata()` option, you can also access those variables in a new dataset, for example, to show the set frequencies as a bar chart. The result is not shown here, but some indicative syntax follows.

```
. tempfile schnable
. upsetplot Arabidopsis Rice Maize Sorghum [fw=freq], varlabels
> gsort(-_degree -_freq) `paperopts' baropts(`toptitle') savedata("`schnable'")
```

(*output omitted*)

```
. use "`schnable'", clear
. graph hbar (asis) _setfreq, over(_set, sort(1)) blabel(bar) ysc(off) `toptitle'
```

3.2 Missing data patterns example

We turn to looking at the patterns of missing values in one of Stata's datasets. Here `upsetplot` is a graphical parallel to `misstable patterns`, but it is a parallel that you may find more readable.

```
. webuse nlswork, clear
(National Longitudinal Survey of Young Women, 14-24 years old in 1968)
```

After running the `missings` command (Cox [2015]; `search dm0085, entry` in an up-to-date Stata to find the latest version), we decide to look at the structure of missing values in five variables. To apply `upsetplot`, we need indicator variables with values 1 if missing and 0 otherwise.

```
. foreach v in ind_code union wks_ue tenure wks_work {
  2.     generate M`v' = missing(`v')
  3.     label variable M`v' "`v'"
  4. }
```

As before, we use a convenient overall title and choose a gray color for the legend markers. More subsets are shown in these plots than in previous examples, so we need to squeeze the bar label text size a little.

```
. local toptitle "t1title(Number of missing values)"
. local paperopts "matrixopts(mc(gs8 ..))"
. upsetplot M*, varlabels baropts(`toptitle') labelopts(mlabsize(vsmall))
> `paperopts'
```

(*output omitted*)

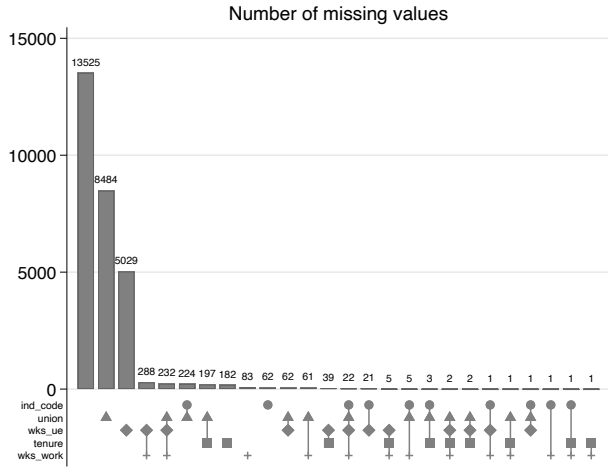

Figure 4. `upsetplot` showing the structure of missingness in five variables from Stata dataset `nlswork.dta`

Figure 4 includes the subset in which none of the variables mentioned are missing, for which `_binary` is 00000 or 0. We can suppress that subset with an extra `if` qualifier, which is satisfied if any (one or more) of the named variables is missing and hence not satisfied if none of them are. Note the close resemblance to the tabular version, which would be shown by the following command:

```
. misstable patterns ind_code union wks_ue tenure wks_work
```

In the variant in figure 5, emphasis is placed on numbers of missing values both across observations and across variables.

```
. upsetplot M* if missing(ind_code, union, wks_ue, tenure, wks_work), varlabels
> baropts(`toptitle') labelopts(mlabsize(vsmall)) `paperopts'
> gsort(_degree -_freq)
```

(*output omitted*)

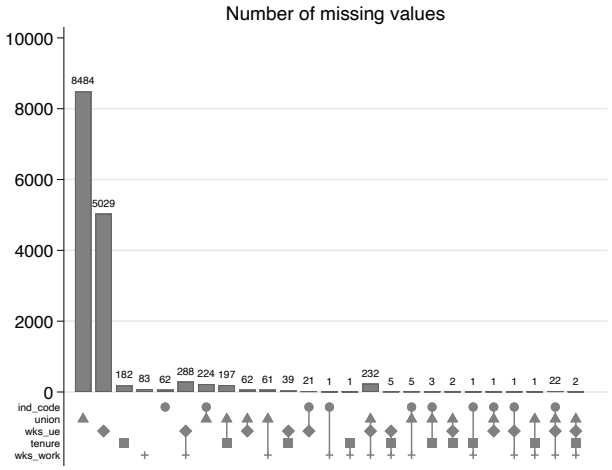

Figure 5. `upsetplot` showing the structure of missingness in five variables from Stata dataset `nlswork.dta`. Observations without any missing values on those variables are omitted. Bars are ordered first by number of missing values in each observation and then by number of observations in each subset.

3.3 Recasting from bars to droplines or spikes

Section 2.2 mentioned it is possible to show droplines or spikes rather than bars. Because the same information is shown, that choice would be a matter of taste or judgment. The syntax for recasting would be `baropts(recast(dropline))` or `baropts(recast(spike))`.

4 vennbar

For consistency, `vennbar` might have been called, say, `vennbarchart`, with "Venn bar chart" being a little like "upsetplot", but that would have been too long and tedious to type.

More importantly, much flexibility comes free with use of `graph hbar`, particularly good support for groupings from `over()` options. Note that `blabel(bar)` is among the defaults; it can be undone with `blabel(none)`.

As in general, using `graph hbar` is a good idea to ensure that text is always easy to read or as easy to read as possible. It remains a little surprising how many users choose `graph bar` but then struggle with a need to reduce axis-label text size or place axis labels at an awkward angle. But you can call up `graph bar` or `graph dot` if you prefer.

We will revisit the datasets examined in section 3.

4.1 Genomics example

We are optimistic that you `save`d the dataset used in section 3.1. A local macro used there can be quickly redefined if needed.

```
. use arms, clear
. local toptitle "t1title(Number of gene families)"
```

Figure 6 is close to defaults and shows subsets in decreasing frequency order, reading from top to bottom. In detail, it owes a little to earlier experiments that showed that the y axis (which here is horizontal) should be extended to accommodate the bar label for the most frequent bar. You may agree with the suggestion detailed in Cox (2012) that when graphs have table flavor, the text for the horizontal axis often looks better at the top.

```
. vennbar Arabidopsis Rice Maize Sorghum [fw=freq], `toptitle' varlabels
> ysc(alt) ysc(r(. 9200))
```

(*output omitted*)

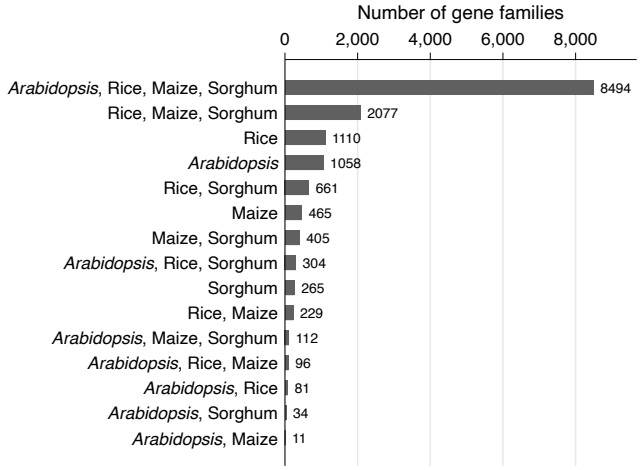

Figure 6. Venn bar chart for overlapping gene families. By default, subsets are ordered top to bottom by frequency.

By the way, a detail that is puzzling until it is familiar is that with `graph hbar`, `graph bar`, and `graph dot` one axis is considered to be a categorical axis, while the other axis, which shows whatever statistic varies over those categories, is always considered to be the y axis, even if it is horizontal. That terminology flouts mathematical convention, but it is matched by the graph syntax. Its rationale is that you can flip between horizontal bar charts and vertical bar charts (which some call column charts) just by changing the command between `graph hbar` and `graph bar`. `graph dot` does not follow suit exactly: it is by default horizontal, but there is an undocumented `vertical` option.

As already mentioned, the bar chart can be recast as a dot chart if that is preferred. The graph is not shown here, but the syntax may be of interest. The rendering of grid lines follows personal preferences for solid but thin lines. `blabel(bar)` continues to work, which may come as a surprise. It is not documented for `graph dot`.

```
. vennbar Arabidopsis Rice Maize Sorghum [fw=freq], `toptitle' varlabels
> sep("; ") ysc(r(0 10000)) recast(dot) linetype(line) lines(lc(gs8) lw(thin))
```

(*output omitted*)

Note that the `recast()` option, which is a standard `twoway` option, is peculiar to `vennbar` and not an option of `graph hbar`, `graph bar`, or `graph dot`. Its name was inspired by the `twoway` option with the same name. The motive is the same: to allow easy experimentation with different graph forms.

As with `upsetplot`, `vennbar` offers a great deal of flexibility over sorting of bars (or dots). Here the flexibility is not offered through a specific `gsort` option but through the machinery of `over()` options.

Figure 7 is an example of a different sort order. A way to remember what happens with multiple `over()` options is that the outermost (last) `over()` option in the syntax determines the outermost (primary) grouping. Hence, the ordering is first on `_degree` and then on `_text`, except that the latter is sorted on `_decimal`.

```
. vennbar Arabidopsis Rice Maize Sorghum [fw=freq],
> over(_text, sort(_freq) descending) over(_degree, descending) nofill
> `toptitle' ysc(alt range(. 9200))
```

(*output omitted*)

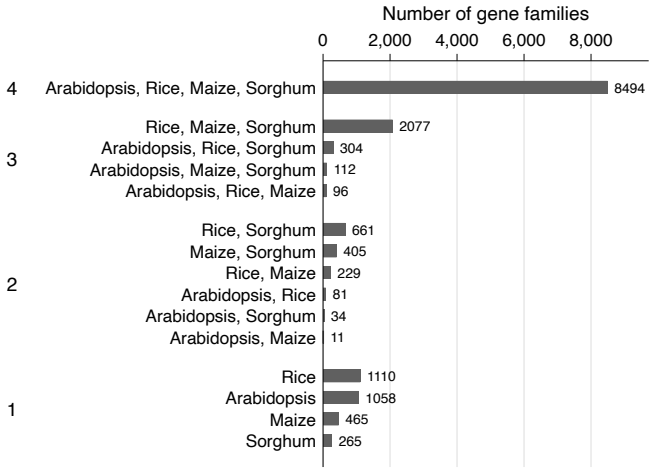

Figure 7. Venn bar chart for overlapping gene families. The ordering is by degree and decimal code, but text is shown.

Although the graph is not shown here, it is easy (for example) to flip sort order for degree to the opposite direction by removing `descending`. Note that because ascending order is the default, `ascending` is not a suboption.

```
. vennbar Arabidopsis Rice Maize Sorghum [fw=freq],
> over(_degree, descending) over(_text, sort(_freq) descending) nofill
> `toptitle' ysc(alt)
```

4.2 Missing-data example

Now we revisit the missing-data example using `nlswork.dta`. Once again, after the data are read in, we create indicator variables for missing values and use a local macro for the graph `toptitle`.

```
. webuse nlswork, clear
(National Longitudinal Survey of Young Women, 14-24 years old in 1968)
. foreach v in ind_code union wks_ue tenure wks_work {
  2. generate M`v' = missing(`v')
  3. label variable M`v' "`v'"
  4. }
. local toptitle  "t1title(Number of missing values)"
```

Figure 8 is close to the default except for those options specified. In such cases, whatever you do can be right or wrong for a given context, such as using `select()` in `upsetplot` to show only the most common subsets or, as here, insisting on a display of subset frequencies for all bars. Sometimes, one plot is helpful for exploration, but another is preferable for presentation.


```
. webuse nlswork, clear
(National Longitudinal Survey of Young Women, 14-24 years old in 1968)
. foreach v in ind_code union wks_ue tenure wks_work {
  2. generate M`v' = missing(`v')
  3. label variable M`v' "`v'"
  4. }
. local toptitle  "t1title(Number of missing values)"
```


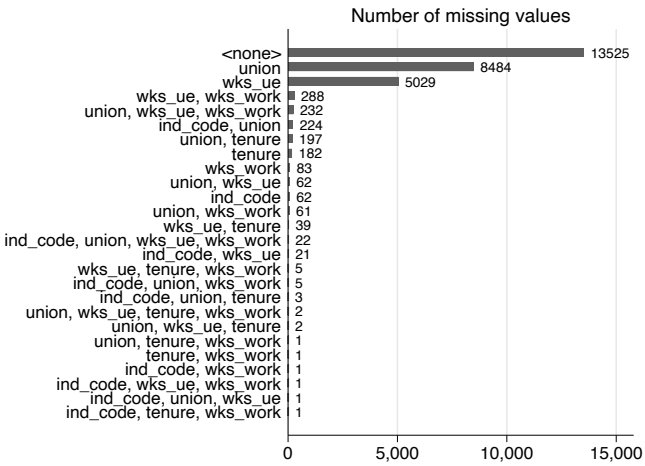

Figure 8. Venn bar chart showing the structure of missingness in five variables from Stata dataset `nlswork.dta`

To reduce the range of frequencies shown, we can omit observations with no missing values on any of the variables mentioned. Often, a focus on how many missing values there are is helpful as a measure of the problem to be faced and to understand the missingness patterns, namely, which variables tend to be observed or missing together. Figure 9 is our final plot example.

```
. vennbar M* if missing(ind_code, union, wks_ue, tenure, wks_work), `toptitle'
> varlabels over(_text, sort(_freq) descending) over(_degree) nofill
> ysc(r(. 9200))
```

(*output omitted*)

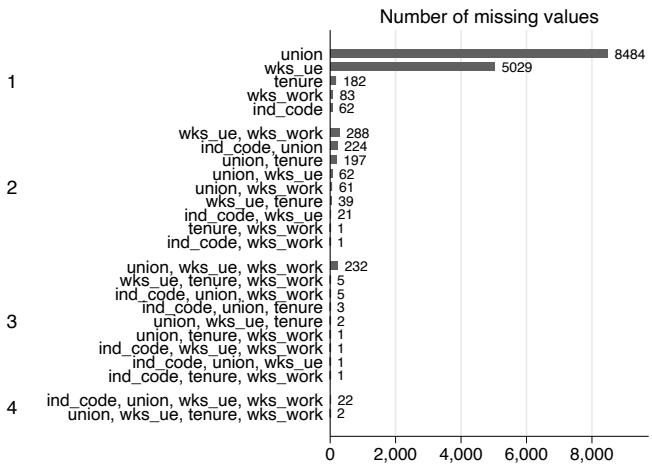

Figure 9. Venn bar chart showing the structure of missingness in five variables from Stata dataset `nlswork.dta`. Observations without any missing values on those variables are omitted. Bars are ordered first by number of missing values in each observation and then by number of observations in each subset.

We close these examples with a reminder that if you prefer table form over such graphs, then reach for the `savedata()` option, which gives access to the numerical results as a Stata dataset. This applies to `upsetplot` as much as to `vennbar`. For missing data, this offers more flexibility than that provided by `misstable patterns` (such as frequencies instead of percentages).

5 Syntax of upsetplot

`upsetplot` *varlist* [*if*] [*in*] [*weight*] [`, fillin percent frformat(`*str*`)`
`pcformat(`*str*`) select(`*numlist*`) varlabels separator(`*str*`) gsort(`*str*`)`
`axisgap(`*str*`) vargap(`*str*`) baropts(`*str*`) labelopts(`*str*`) matrixopts(`*str*`)`
`spikeopts(`*str*`)` *graph_options* `savedata(`*filespec*`)`]

`fweight`s and `aweight`s are allowed; see [U] **11.1.6 weight**.

5.1 Description

By default, `upsetplot` produces bar chart alternatives to Euler or Venn diagrams showing the frequencies (meaning generally, abundances) of subsets of observations as defined jointly by a bundle of numeric indicator variables. Such plots resemble what have been called UpSetPlots elsewhere. There is scope to recast results to use some other subcommands of `twoway`.

The order of variables presented to `upsetplot` does not determine the order in which they are shown in a plot. By default, bars (or spikes or dotted lines) are shown in order of subset frequency, but choosing a more suitable order is the user's prerogative. There is considerable scope to change that sort order using other criteria.

Commonly, but not necessarily, subset frequencies (abundances) are already in a variable in the dataset. If so, that variable should be specified as frequency or analytic weights. If no weights are specified, `upsetplot` counts observations for you. Either way, note that the focus of this command is on displaying frequencies and not on the particular values in each subset.

The display uses various subcommands of `graph twoway`. The most distinctive feature is a matrix- or table-like legend below the bar chart with marker symbols shown on each row whenever any indicator is 1 in a subset. Correspondingly, absence of a marker symbol implies that indicator is 0 in that subset. Each row is labeled by a variable name or optionally by a variable label. In either case, variable name or variable label, short text is desirable. Vertical spikes connecting uppermost and lowermost marker symbols are added.

The reduced dataset used by `upsetplot` may be saved for future work using the `savedata()` option. This dataset may be as useful as or more useful than the plot. Saving results allows greater flexibility in plotting. Tabulation or other reporting is also made easier. Results often need to be scaled in some way, for example, by looking at conditional proportions or percents. (Optionally, percents can be saved too.)

The variables in such a reduced dataset are the original indicator variables and as follows. The names here may thus not be used as names for the indicator variables specified.

`_binary` is a string variable containing a binary code such as `"00"`, `"01"`, `"10"`, or `"11"`.

`_decimal` is a numeric variable containing a decimal equivalent such as `0`, `1`, `2`, or `3`.

`_text` is a string variable containing a description of each subset using variable names or optionally variable labels. The text `"<none>"` is reported for any subset that would otherwise have empty text.

`_freq` is a numeric variable containing the frequency of occurrence of each subset. If analytic weights were specified, values may have fractional parts. Otherwise, values will be integer counts.

(Optionally) `_percent` is a numeric variable containing the percent occurrence of each subset.

`_degree` is a numeric variable indicating the degree of each subset (number of participating sets), counted as true (1) according to each indicator variable.

`_set` is a string variable that indicates each set using its variable name or optionally its variable label.

`_setfreq` is a numeric variable that indicates the frequency of each set.

`_set` and `_setfreq` are physically but not logically aligned with the other variables mentioned above.

5.2 Options

5.2.1 What to show

`fillin` insists on showing subsets that do not occur with their frequency zero. This can be helpful if there are only a few such subsets, but it is not usually helpful otherwise.

`percent` specifies listing and plotting of percents rather than frequencies.

`frformat(`*str*`)` specifies a display format for frequencies in listings. This option may be appropriate if any frequencies include fractional parts. If you wish to specify a format for the bar labels, you should do so directly using `labelopts(mlabformat())`.

`pcformat(`*str*`)` specifies a display format for percents in listings and plots. The default is `pcformat(%2.1f)`. This option has no effect without `percent`. If you wish to specify a format for the bar labels, do so directly using `labelopts(mlabformat())`.

`select(`*numlist*`)` specifies that only the first so many bars be shown. That is, without this option, the bars that would be shown are considered to be numbered 1, 2, 3, and so on, from left to right. `select(1/10)` would reduce the display to the first 10 bars. Typically, this option would be used to select the most frequent subsets, but the syntax allows any integer *numlist*.

5.2.2 Detail of display

`varlabels` specifies use of variable labels to describe each subset. The default is to use variable names. In either case, only variables taking on value 1 in each subset are named or labeled. If a variable label has not been defined, the variable name is used instead.

`separator(`*str*`)` specifies a string to separate variable names or, as above, variable labels in calculating `_text`. The default is `", "`, a comma followed by a space. Hint: The intersection symbol can be obtained using `"{&cap}"` or Unicode character (U+2229) through `uchar(2229)`. The Stata Markup and Control Language (SMCL) notation will be interpreted on graphs but will appear uninterpreted in data listings. The Unicode character should be interpreted in both. This option has no bearing on the graph produced by `upsetplot` but is explained here for symmetry with the help for `vennbar`.

`gsort(`*str*`)` specifies instructions to `gsort` on the order of bars, mentioning one or more variable names for sorting the display. The variables named must be included in the reduced dataset as defined above and could be one or more of the following: `_binary`, `_decimal`, `_text`, `_freq`, `_percent` (if specified), and `_degree`.

The default is to sort on the frequency (abundance) variable `_freq` created by the command, highest values first.

`axisgap(`*str*`)` specifies a gap between the x axis and the first row of the legend. The default is (the maximum value of `_freq` or `_percent`)/100. After the command has run, the value used is accessible as local macro `axisgap`. That allows one or more extra passes to change the gap.

`vargap(`*str*`)` specifies the gap between each row in the legend. The default is (the maximum value of `_count` or `_percent`)/25. After the command has run, the value used is accessible as local macro `vargap`. That allows one or more extra passes to change the gap.

`baropts(`*str*`)` are options of `twoway bar` used to tune the rendering of bars. The defaults include `baropts(yla(, ang(h)) barw(0.8) xsc(off))`.

`labelopts(`*str*`)` are options of `twoway scatter` used to tune the rendering of text labels showing frequencies or percents above each bar. The defaults are `labelopts(mla(_freq))` or `labelopts(mla(_percent))` with `ms(none) mlabc(black) mlabpos(12) mlabsize(small)`. Alternatively, `labelopts(none)` suppresses such labels.

`matrixopts(`*str*`)` are options of `twoway scatter` used to tune the matrix- or tablelike legend. The defaults include `matrixopts(ymla(, ang(h) noticks) legend(off) aspect(0.8) ms(O T D S + X))` and marker colors as defined by Okabe and Ito (2002) (on which see, for example, Wong [2011] or Wilke [2019]).

Note that something like `matrixopts(ms(O ..) mc(gs8 ..))` would get you closer to conforming with a widespread if unimaginative convention.

`spikeopts(`*str*`)` are options of `twoway rspike` used to tune the spikes connecting markers in the legend. The default is `spikeopts(lc(gs8))`.

graph_options are other options of `graph`. An example might be `name()`. Note that `graph` may not be especially smart about any space needed above the highest bar label, so you may need two passes and a call to `yscale()` to extend the axis.

5.2.3 Saving results as new dataset

`savedata(`*filespec*`)` specifies a (filepath and) filename for saving results to a new dataset. The specification may include `, replace`—which is needed to replace any existing dataset with the same path and name.

6 Syntax of vennbar

`vennbar` *varlist* [*if*] [*in*] [*weight*] [`, fillin percent frformat(`*str*`)` `pcformat(`*str*`) varlabels vallabels separator(`*str*`) recast(`*subcmd*`)` *graph_options* `savedata(`*filespec*`)`]

`fweight`s and `aweight`s are allowed; see [U] **11.1.6 weight**.

6.1 Description

`vennbar` produces bar or dot chart alternatives to Euler or Venn diagrams showing the frequencies (meaning generally, abundances) of subsets of observations as defined jointly by a bundle of numeric indicator variables.

The order of variables presented to `vennbar` does not determine the order in which they are shown in a plot. By default, bars (or dotted lines) are shown in order of subset frequency, but choosing a more suitable order is the user's prerogative. There is considerable scope to change that sort order using other criteria.

Commonly, but not necessarily, subset frequencies (abundances) are already in a variable in the dataset. If so, that variable should be specified as frequency or analytic weights. If no weights are specified, `vennbar` counts observations for you. Either way, note that the focus of this command is on displaying frequencies and not on the particular values in each subset.

The display by default uses `graph hbar` but may optionally be recast as using `graph bar` (which is not usually advised) or as using `graph dot` (which may appeal). The choice is a matter of personal taste, although in general horizontal displays make it easier to show and read values or labels of subsets.

The reduced dataset used by `vennbar` may be saved for future work using the `savedata()` option. This dataset may be as useful as or more useful than the plot. Saving results allows greater flexibility in plotting. Tabulation or other reporting is also made easier. Results often need to be scaled in some way, for example, by looking at conditional proportions or percents. (Optionally, percents can be saved too.)

The variables in such a reduced dataset are the original indicator variables and as follows. Thus, the names here may not be used as names for the indicator variables specified.

`_binary` is a string variable containing a binary code such as `"00"`, `"01"`, `"10"`, or `"11"`.

`_decimal` is a numeric variable containing a decimal equivalent such as `0`, `1`, `2`, or `3`.

`_text` is a string variable containing a description of each subset using variable names, variable labels, or value labels. The text `"<none>"` is reported for any subset which would otherwise have empty text.

`_freq` is a numeric variable containing the frequency of occurrence of each subset. If analytic weights were specified, values may have fractional parts. Otherwise, values will be integer counts.

(Optionally) `_percent` is a numeric variable containing the percent occurrence of each subset.

`_degree` is a numeric variable indicating the degree of each subset (number of participating sets), counted as true (1) according to each indicator variable.

`_set` is a string variable that indicates each set using its variable name or optionally its variable label.

`_setfreq` is a numeric variable that indicates the frequency of each set.

`_set` and `_setfreq` are physically but not logically aligned with the other variables mentioned above.

6.2 Options

6.2.1 What to show

`fillin` insists on showing subsets that do not occur with their frequency zero. This can be helpful if there are only a few such subsets, but it is not usually helpful otherwise.

`percent` specifies listing and plotting of percents rather than frequencies.

`frformat(`*str*`)` specifies a display format for frequencies in listings. This option may be appropriate if any frequencies include fractional parts. If you wish to specify a format to `blabel()`, you should do so directly; see the help on `blabel()`.

`pcformat(`*str*`)` specifies a display format for percents in listings. The default is `%2.1f`. This option has no effect without `percent`. If you wish to specify a format to `blabel()`, you should do so directly; see the help on `blabel()`.

6.2.2 Detail of display

`varlabels` specifies use of variable labels to describe each subset. The default is to use variable names. In either case, only variables taking on value 1 in each subset are named or labeled. If a variable label has not been defined, the variable name is used instead.

`vallabels` specifies use of value labels to describe each subset. If value labels are not defined, values 0 or 1 will be used instead. With this option, subsets defined by values 0 or 1 will always be labeled somehow. This option may be useful when 0 and 1 represent values that are both of direct interest, such as alive and dead, wet and dry, or female and male.

`separator(`*str*`)` specifies a string to separate variable names or, as above, variable or value labels in display of subsets. The default is `", "`, a comma followed by a space. Hint: The intersection symbol can be obtained using SMCL's `"{&cap}"` or Unicode character (U+2229) through `uchar(2229)`. The SMCL notation will be interpreted on graphs but will appear uninterpreted in data listings. The Unicode character should be interpreted in both.

`recast(`*subcmd*`)` specifies a subcommand of `graph`, either `bar` or `dot`, as an alternative to the default `hbar`. Note: This option name is inspired by the `recast()` option of `twoway` but is not that option. If you wish to use `twoway` instead, specify the `savedata()` option, and fire up `twoway` directly on the results dataset.

graph_options refer to other options of `graph hbar`, `graph bar`, or `graph dot`. As the plot here has table flavor, some of the ideas covered by Cox (2008, 2012) may be helpful. Note that `graph` may not be especially smart about any space needed above the highest bar label, so you may need two passes and a call to `yscale()` to extend the axis.

The default is `over(_text, sort(_count) descending) blabel(bar)`. Otherwise, options may refer to variables included in the reduced dataset as defined above, which could be any of the following: `_binary`, `_decimal`, `_text`, `_freq`, `_percent` (if specified), and `_degree`.

Note that any other `over()` option overrides this default. Thus, if you want that default and other choices too, you must spell out all your choices.

Using one or more `over()` options is often the key to a successful plot. If these options are unfamiliar to you, do study the examples, and check out the help for `graph hbar` for its syntax, its suboptions, and the linked `nofill` option.

6.2.3 Saving results as new dataset

`savedata(`*filespec*`)` specifies a (filepath and) filename for saving results to a new dataset. The specification may include `, replace`—which is needed to replace any existing dataset with the same path and name.

7 Historical remarks and literature survey

The elementary but fundamental idea of representing true (or present) as 1 and false (or absent) as 0 has a splendid history. Although it has yet longer roots, the idea was strongly developed by George Boole (1815–1864): Boole (1854) was his major work in

this territory, on which see particularly Grattan-Guinness (2005). Boole has been given a full-length biography (MacHale 2014) and an even longer sequel (MacHale and Cohen 2018). For shorter accounts, see Gardner (1969; 1979, chap. 8), Broadbent (1970), MacHale (2000, 2008), Heath and Seneta (2001), or Grattan-Guinness (2004). Dewdney (1993) and Gregg (1998) provide examples of how such Boolean algebra features in computing. Knuth (1998, chap. 4.1) gives an excellent historical summary of positional number systems, and Knuth (2011) gives a masterly synopsis, including historical material, of related combinatorial algorithms. Strickland and Lewis (2022) focus on binary arithmetic and logic in the work of Leibniz (1646–1716). The leading biography of Leibniz is by Antognazza (2009), although the earlier biography by Aiton (1985) is still informative. Leibniz's projects feature in many subplots in Stephenson (2003) and its sequels. Cox (2016) makes further Stata-related comments on truth, falsity, and indication. Cox and Schechter (2019) survey the creation of indicator variables in Stata.

Various commentators, from Leibniz onward, have seen anticipations of binary arithmetic in the divination manual *I Ching* (*Yijing*, *Yi Jing*, *Yi King*, etc.). That seems exaggerated. See Gardner (1974; 1986, chap. 20) for a brisk discussion and Knuth (2011) and Strickland and Lewis (2022) for further comments.

The `upsetplot` command was stimulated by several articles since 2014, including Lex (2021, 2022), Lex and Gehlenborg (2014), Lex et al. (2014), Conway, Lex, and Gehlenborg (2017), and Ballarini et al. (2020). In contrast, the `vennbar` command is just an alternative as a more orthodox bar chart.

Euler–Venn diagrams are widely familiar in mathematics and science and indeed as a cultural meme echoed in cartoons, T-shirt or mug designs, and much else. Christianson (2012) mentioned Venn diagrams as one of *100 Diagrams That Changed the World*. We could add, but rather should subtract, examples of diagrams that mimic their form but do not match their logic. Some entertaining examples are given by Bergstrom and West (2020, 149–151). Friendly introductions to set theory featuring Euler–Venn diagrams include Stewart (1975) and Gullberg (1997). Conversely, compare Hamming (1985, 367): "Set theory has been taught until the typical student is weary of it, so we will assume that it is familiar." Beyond their original and continuing use in logic, such diagrams are commonly used in introductions to probability: see (for example) Pitman (1993), Whittle (2000), Dekking et al. (2005), Miller (2017), or Blitzstein and Hwang (2019). Historically and to the present, set theory is linked to much fundamental work in logic, number theory, and other parts of mathematics (Bagaria [2008]; various chapters in Grattan-Guinness [1994]; Stillwell [2010]).

For the history of Euler–Venn and related diagrams, see Baron (1969), Gardner (1982), Edwards (2004), Moktefi and Shin (2012), and Bennett (2015). Friendly and Wainer (2021, 102–103) flag the use of a similar area-proportional diagram by Playfair (1801, opp.p.48). Wilkinson (2012) covers some more recent work on drawing area-proportional plots from a statistical point of view. Macfarlane (1885, 1891) referred to composite categories laid out in sequence as the logical spectrum.

Venn (1880c,a,b, 1881, 1894) made explicit that the diagrams later often named after him grew out of earlier work. Indeed, few logicians were as fully aware of previous

contributions. Thus, the name *Venn diagram* exemplifies Stigler's Law (1980, 1999) that "[n]o scientific discovery is named after its original discoverer". The injustice is partially corrected by crediting Euler's earlier work (1768), on which see conveniently Sandifer (2007) or Bennett (2015). A distinction is often drawn (for example, Mollerup [2015, 166]) that Venn diagrams show all possible combinations, while Euler diagrams show only actual combinations. However, Euler's contribution in turn was preceded by yet earlier work by Leibniz and several other scholars. Nevertheless, crediting Euler, Venn, or both is fair and there is no point to suggesting yet another term.

John Venn (1834–1923) now benefits from a full-length biography, Verburgt (2022). For shorter appreciations, see Broadbent (1976), Grattan-Guinness (2001), or Gibbins (2004). Grattan-Guinness (2011) places the work of Boole and Venn in context, surveying the development of logic in 19th century Britain. Venn's interest in probability and statistics was profound: see especially his first book *The Logic of Chance* (1866, 1876, 1888) and a still useful review article on averages (Venn 1891).

Leonhard Euler (1707–1783) is also well served by a full-length biography (Calinger 2016). See also Calinger, Denisova, and Polyakhova (2019) on what in English is known as *Letters to a German Princess*. For a concise overview of some of his mathematical achievements, see Dunham (1999). For a shorter although still detailed account, see Youschkevitch (1971). For a very concise account, see Sandifer (2008).

8 Conclusion

This article takes as given that—beyond simple examples—annotated Euler–Venn diagrams have severe limitations in showing subset frequencies, even when cleverly drawn. They may allow individual detail to be read off easily, but they can be poor at showing the big picture of the distribution from common to rare subsets and do not offer any visualization of frequencies (or more generally abundances).

Whatever the graph preferred (or even if tables are desired), you need a data structure showing the information on overlap. That is calculated from a bundle of indicator variables—and, quite possibly, an already existing variable containing abundances.

The main contribution of this article is to document two new commands, `upsetplot` and `vennbar`. Initial experience of our own and reactions from others have already underlined what is standard in visualization, namely, that different people prefer different designs. We offer both for the collective Stata toolbox.

9 Acknowledgments

This article arose from a conversation between the authors at the London Stata meeting in 2022. We much appreciate the enterprise, energy, and enthusiasm behind these meetings over almost 30 years shown by the late Ana Timberlake, Teresa Timberlake, David Corbett, and their colleagues.

Angela Wood originally drew the attention of Tim P. Morris to upsetplots. Antony Unwin provided Nicholas J. Cox access to his forthcoming book. Several posts on Statalist indicated very helpfully both interest in this problem and reactions to earlier versions of these commands.

10 Programs and supplemental materials

To install a snapshot of the corresponding software files as they existed at the time of publication of this article, type

```
. net sj 24-2
. net install gr0095     (to install program files, if available)
. net get gr0095         (to install ancillary files, if available)
```

11 References

Aiton, E. J. 1985. *Leibniz: A Biography*. Bristol, U.K.: Adam Hilger.

Allenby, R. B. J. T., and A. Slomson. 2011. *How to Count: An Introduction to Combinatorics*. Boca Raton, FL: CRC Press.

Antognazza, M. R. 2009. *Leibniz: An Intellectual Biography*. Cambridge: Cambridge University Press.

Bagaria, J. 2008. Set theory. In *The Princeton Companion to Mathematics*, ed. T. Gowers, J. Barrow-Green, and I. Leader, 615–634. Princeton, NJ: Princeton University Press.

Ballarini, N. M., Y.-D. Chiu, F. König, M. Posch, and T. Jaki. 2020. A critical review of graphics for subgroup analyses in clinical trials. *Pharmaceutical Statistics* 19: 541–560. https://doi.org/10.1002/pst.2012.

Baron, M. E. 1969. A note on the historical development of logic diagrams: Leibniz, Euler and Venn. *Mathematical Gazette* 53: 113–125. https://doi.org/10.2307/3614533.

Bennett, D. 2015. Origins of the Venn diagram. In *Research in History and Philosophy of Mathematics*, ed. M. Zack and E. Landry, 105–119. Cham, Switzerland: Springer. https://doi.org/10.1007/978-3-319-22258-5_8.

Bergstrom, C. T., and J. D. West. 2020. *Calling Bullshit: The Art of Skepticism in a Data-Driven World*. New York: Random House.

Blitzstein, J. K., and J. Hwang. 2019. *Introduction to Probability*. 2nd ed. Boca Raton, FL: CRC Press.

Boole, G. 1854. *An Investigation of the Laws of Thought, on Which are Founded the Mathematical Theories of Logic and Probabilities*. London: Walton and Maberley.

Broadbent, T. A. A. 1970. Boole, George. In Vol. 2 of *Dictionary of Scientific Biography*, ed. C. C. Gillispie, 293–298. New York: Charles Scribner's Sons.

———. 1976. Venn, John. In Vol. 13 of *Dictionary of Scientific Biography*, ed. C. C. Gillispie, 611–613. New York: Charles Scribner's Sons.

Calinger, R. S. 2016. *Leonhard Euler: Mathematical Genius in the Enlightenment*. Princeton, NJ: Princeton University Press. https://doi.org/10.1515/9781400866632.

Calinger, R. S., E. Denisova, and E. N. Polyakhova. 2019. *Leonhard Euler's Letters to a German Princess: A Milestone in the History of Physics Textbooks and More*. San Rafael, CA: Morgan and Claypool Publishers. https://doi.org/10.1088/2053-2571/aae6d2.

Christianson, S. 2012. *100 Diagrams That Changed the World: From the Earliest Cave Paintings to the Innovation of the iPod*. New York: Penguin.

Conway, J. R., A. Lex, and N. Gehlenborg. 2017. UpSetR: An R package for the visualization of intersecting sets and their properties. *Bioinformatics* 33: 2938–2940. https://doi.org/10.1093/bioinformatics/btx364.

Cox, N. J. 2007. Stata tip 52: Generating composite categorical variables. *Stata Journal* 7: 582–583. https://doi.org/10.1177/1536867X0800700407.

———. 2008. Speaking Stata: Between tables and graphs. *Stata Journal* 8: 269–289. https://doi.org/10.1177/1536867X0800800208.

———. 2010. Speaking Stata: Finding variables. *Stata Journal* 10: 281–296. https://doi.org/10.1177/1536867X1001000208.

———. 2012. Speaking Stata: Axis practice, or what goes where on a graph. *Stata Journal* 12: 549–561. https://doi.org/10.1177/1536867X1201200314.

———. 2015. Speaking Stata: A set of utilities for managing missing values. *Stata Journal* 15: 1174–1185. https://doi.org/10.1177/1536867X1501500413.

———. 2016. Speaking Stata: Truth, falsity, indication, and negation. *Stata Journal* 16: 229–236. https://doi.org/10.1177/1536867X1601600117.

———. 2017. Speaking Stata: Tables as lists: The groups command. *Stata Journal* 17: 760–773. https://doi.org/10.1177/1536867X1701700314.

———. 2018. Software Updates: Tables as lists: The groups command. *Stata Journal* 18: 291. https://doi.org/10.1177/1536867X1801800118.

Cox, N. J., and C. B. Schechter. 2019. Speaking Stata: How best to generate indicator or dummy variables. *Stata Journal* 19: 246–259. https://doi.org/10.1177/1536867X19830921.

Dekking, F. M., C. Kraikamp, H. P. Lopuhaä, and L. E. Meester. 2005. *A Modern Introduction to Probability and Statistics: Understanding Why and How*. London: Springer. https://doi.org/10.1007/1-84628-168-7.

Dewdney, A. K. 1993. *The New Turing Omnibus: 66 Excursions in Computer Science*. New York: Henry Holt.

D'Hont, A., F. Denoeud, J.-M. Aury, F.-C. Baurens, F. Carreel, O. Garsmeur, B. Noel, et al. 2012. The banana (*Musa acuminata*) genome and the evolution of monocotyledonous plants. *Nature* 488: 213–217. https://doi.org/10.1038/nature11241.

Dunham, W. 1999. *Euler: The Master of Us All*. Washington, DC: Mathematical Association of America.

Edwards, A. W. F. 2004. *Cogwheels of the Mind: The Story of Venn Diagrams*. Baltimore: Johns Hopkins University Press.

Euler, L. 1768. Vol. 2 of *Lettres à Une Princesse d'Allemagne sur Divers Sujets de Physique et de Philosophie*. Paris: Saint Petersbourg: L'Académie Impériale des Sciences. https://doi.org/10.5962/bhl.title.16687.

Freeman, S. C., E. Saeedi, J. M. Ordìñez-Mena, C. R. Nevill, J. Hartmann-Boyce, D. M. Caldwell, N. J. Welton, N. J. Cooper, and A. J. Sutton. 2023. Data visualisation approaches for component network meta-analysis: Visualising the data structure. *BMC Medical Research Methodology* 23: Article 208. https://doi.org/10.1186/s12874-023-02026-z.

Friendly, M., and H. Wainer. 2021. *A History of Data Visualization and Graphic Communication*. Cambridge, MA: Harvard University Press.

Gardner, M. 1969. Mathematical games: Boolean algebra, Venn diagrams and the propositional calculus. *Scientific American* 220(2): 110–114. https://doi.org/10.1038/scientificamerican0269-110.

———. 1974. Mathematical games: The combinatorial basis of the "I Ching," the Chinese book of divination and wisdom. *Scientific American* 230(1): 108–113. https://doi.org/10.1038/scientificamerican0174-108.

———. 1979. *Mathematical Circus*. New York: Alfred A. Knopf.

———. 1982. *Logic Machines and Diagrams*. 2nd ed. Chicago: University of Chicago Press.

———. 1986. *Knotted Doughnuts and other Mathematical Entertainments*. New York: Freeman.

Gibbins, J. R. 2004. Venn, John. In Vol. 56 of *Oxford Dictionary of National Biography*, ed. H. C. G. Matthew and B. Harrison, 259–260. Oxford: Oxford University Press. https://doi.org/10.1093/ref:odnb/36639.

Gleason, A. M. 1991. *Fundamentals of Abstract Analysis*. Boston: Jones and Bartlett.

Gong, W., and J. Ostermann. 2011. pvenn: Stata module to create proportional Venn diagram. Statistical Software Components S457368, Department of Economics, Boston College. https://ideas.repec.org/c/boc/bocode/s457368.html.

Grattan-Guinness, I., ed. 1994. *Companion Encyclopedia of the History and Philosophy of the Mathematical Sciences*. London: Routledge.

Grattan-Guinness, I. 2001. John Venn. In *Statisticians of the Centuries*, ed. C. C. Heyde, E. Seneta, P. Crépel, S. E. Fienberg, and J. Gani, 194–196. New York: Springer. https://doi.org/10.1007/978-1-4613-0179-0_40.

———. 2004. Boole, George. In Vol. 6 of *Oxford Dictionary of National Biography*, ed. H. C. G. Matthew and B. Harrison, 582–585. Oxford: Oxford University Press. https://doi.org/10.1093/ref:odnb/2868.

———. 2005. George Boole, *An investigation of the laws of thought on which are founded the mathematical theory of logic and probabilities* (1854). In *Landmark Writings in Western Mathematics 1640–1940*, ed. I. Grattan-Guinness, R. Cooke, L. Corry, P. Crépel, and N. Guicciardini, 470–479. Amsterdam: Elsevier. https://doi.org/10.1016/B978-044450871-3/50117-0.

———. 2011. Victorian logic: From Whately to Russell. In *Mathematics in Victorian Britain*, ed. R. Flood, A. Rice, and R. Wilson, 359–374. Oxford: Oxford University Press.

Gregg, J. R. 1998. *Ones and Zeros: Understanding Boolean Algebra, Digital Circuits, and the Logic of Sets*. Piscataway, NJ: IEEE.

Gullberg, J. 1997. *Mathematics: From the Birth of Numbers*. New York: W. W. Norton.

Hamming, R. W. 1985. *Methods of Mathematics Applied to Calculus, Probability, and Statistics*. Englewood Cliffs, NJ: Prentice-Hall.

———. 1991. *The Art of Probability for Scientists and Engineers*. Reading, MA: Addison-Wesley.

Heath, P., and E. Seneta. 2001. George Boole. In *Statisticians of the Centuries*, ed. C. C. Heyde, E. Seneta, P. Crépel, S. E. Fienberg, and J. Gani, 167–170. New York: Springer. https://doi.org/10.1007/978-1-4613-0179-0_34.

Knuth, D. E. 1998. *The Art of Computer Programming*. Vol. 2, *Seminumerical Algorithms*. 3rd ed. Reading, MA: Addison–Wesley.

———. 2011. *The Art of Computer Programming*. Vol. 4A, *Combinatorial Algorithms, Part 1*. Upper Saddle River, NJ: Addison–Wesley.

Kosara, R. 2007. Autism diagnosis accuracy—Visualization redesign. https://eagereyes.org/criticism/autism-diagnosis-accuracy.

Lauritsen, J. M. 1999a. gr34: Drawing Venn diagrams. *Stata Technical Bulletin* 47: 3–8. Reprinted in *Stata Technical Bulletin Reprints*. Vol. 8, pp. 65–71. College Station, TX: Stata Press.

———. 1999b. gr34.1: Drawing Venn diagrams. *Stata Technical Bulletin* 48: 2. Reprinted in *Stata Technical Bulletin Reprints*. Vol. 8, pp. 71–72. College Station, TX: Stata Press.

———. 1999c. gr34.2: Drawing Venn diagrams. *Stata Technical Bulletin* 49: 8. Reprinted in *Stata Technical Bulletin Reprints*. Vol. 9, p. 89. College Station, TX: Stata Press.

———. 1999d. gr34.3: An update to drawing Venn diagrams. *Stata Technical Bulletin* 54: 17–19. Reprinted in *Stata Technical Bulletin Reprints*. Vol. 9, pp. 89–92. College Station, TX: Stata Press.

———. 2009. venndiag: Stata module to generate Venn diagrams. Statistical Software Components S361502, Department of Economics, Boston College. https://ideas.repec.org/c/boc/bocode/s361502.html.

Lex, A. 2021. UpSet: Visualizing intersecting sets. https://upset.app/.

———. 2022. "Ah, I missed that. It's not all that meaningful. It comes from me being 'upset' when I saw this chart... And then upset has 'set' in it...". Twitter, September 13, 2022, 12:34 p.m. https://mobile.twitter.com/alexander_lex/status/1569741352417787905.

Lex, A., and N. Gehlenborg. 2014. Sets and intersections. *Nature Methods* 11: 779. https://doi.org/10.1038/nmeth.3033.

Lex, A., N. Gehlenborg, H. Strobelt, R. Vuillemot, and H. Pfister. 2014. UpSet: Visualization of intersecting sets. *IEEE Transactions on Visualization and Computer Graphics* 20: 1983–1992. https://doi.org/10.1109/TVCG.2014.2346248.

Macfarlane, A. 1885. The logical spectrum. *Philosophical Magazine*, 5th ser., 19: 286–290. https://doi.org/10.1080/14786448508627677.

———. 1891. Adaption of the method of the logical spectrum to Boole's problem. In Vol. 39 of *Proceedings of the American Association of the Advancement of Science*, ed. F. W. Putnam, 57–60. Salem, MA: Salem Press Publishing and Printing.

MacHale, D. 2000. George Boole 1815–1864. In *Creators of Mathematics: The Irish Connection*, ed. K. Houston, 27–32. Dublin: University College Dublin Press.

———. 2008. George Boole (1815–1864). In *The Princeton Companion to Mathematics*, ed. T. Gowers, J. Barrow-Green, and I. Leader, 769–770. Princeton, NJ: Princeton University Press.

———. 2014. *The Life and Work of George Boole: A Prelude to the Digital Age*. Cork, Ireland: Cork University Press.

MacHale, D., and Y. Cohen. 2018. *New Light on George Boole*. Cork, Ireland: Cork University Press.

Miller, S. J. 2017. *The Probability Lifesaver: All the Tools You Need to Understand Chance*. Princeton, NJ: Princeton University Press. https://doi.org/10.2307/j.ctvc7767n.

Moktefi, A., and S.-J. Shin. 2012. A history of logic diagrams. In *Handbook of the History of Logic*. Vol. 11, *Logic: A History of Its Central Concepts*, ed. D. M. Gabbay, F. J. Pelletier, and J. Woods, 611–682. Amsterdam: North-Holland. https://doi.org/10.1016/B978-0-444-52937-4.50011-3.

Mollerup, P. 2015. *Data Design: Visualizing Quantities, Locations, Connections*. London: Bloomsbury.

Okabe, M., and K. Ito. 2002. Color universal design (CUD). How to make figures and presentations that are friendly to colorblind people. https://jfly.uni-koeln.de/color/.

Over, M. 2022. pvenn2: Proportional Venn diagram, enhanced version of pvenn. http://digital.cgdev.org/doc/stata/MO/Misc.

Pitman, J. 1993. *Probability*. New York: Springer. https://doi.org/10.1007/978-1-4612-4374-8.

Playfair, W. 1801. *The Statistical Breviary; Shewing, on a Principle Entirely New, the Resources of Every State and Kingdom in Europe*. London: Wallis.

Richards, D. 2023. *The Martin Gardner Bibliography*. Stanford, CA: CSLI Publications.

Sandifer, C. E. 2007. *How Euler Did It*. Washington, DC: Mathematical Association of America.

———. 2008. Leonhard Euler (1707–1783). In *The Princeton Companion to Mathematics*, ed. T. Gowers, J. Barrow-Green, and I. Leader, 747–749. Princeton, NJ: Princeton University Press.

Schnable, P. S., D. Ware, R. S. Fulton, J. C. Stein, F. Wei, S. Pasternak, C. Liang, et al. 2009. The B73 maize genome: Complexity, diversity, and dynamics. *Science* 326: 1112–1115. https://doi.org/10.1126/science.1178534.

Stephenson, N. 2003. *Quicksilver*. Vol. 1, *The Baroque Cycle*. New York: William Morrow.

Stewart, I. 1975. *Concepts of Modern Mathematics*. Harmondsworth: Penguin.

Stigler, S. M. 1980. Stigler's law of eponymy. *Transactions of the New York Academy of Sciences* 39: 147–158. https://doi.org/10.1111/j.2164-0947.1980.tb02775.x.

———. 1999. *Statistics on the Table: The History of Statistical Concepts and Methods*. Cambridge, MA: Harvard University Press. https://doi.org/10.2307/j.ctv1pdrpsj.

Stillwell, J. 2010. *Mathematics and Its History*. 3rd ed. New York: Springer. https://doi.org/10.1007/978-1-4419-6053-5.

Strickland, L., and H. R. Lewis. 2022. *Leibniz on Binary: The Invention of Computer Arithmetic*. Cambridge, MA: MIT Press.

Unwin, A. 2015. *Graphical Data Analysis with R*. Boca Raton, FL: CRC Press.

———. 2024. *Getting (more out of) Graphics: Practice and Principles of Data Visualisation*. Boca Raton, FL: CRC Press.

Venn, J. 1866. *The Logic of Chance*. London: Macmillan.

———. 1876. *The Logic of Chance*. 2nd ed. London: Macmillan.

———. 1880a. On the diagrammatic and mechanical representation of propositions and reasonings. *Philosophical Magazine*, 5th ser., 10(59): 1–18. https://doi.org/10.1080/14786448008626877.

———. 1880b. On the employment of geometrical diagrams for the sensible representation of logical propositions. *Transactions of the Cambridge Philosophical Society* 4: 47–59.

———. 1880c. On the forms of logical proposition. *Mind* 5: 336–349. https://doi.org/10.1093/mind/os-V.19.336.

———. 1881. *Symbolic Logic*. London: Macmillan.

———. 1888. *The Logic of Chance: An Essay on the Foundations and Province of the Theory of Probability, with Especial Reference to its Logical Bearings and its Application to Moral and Social Science and to Statistics*. 3rd ed. London: Macmillan.

———. 1891. On the nature and uses of averages. *Journal of the Royal Statistical Society* 54: 429–456. https://doi.org/10.2307/2979569.

———. 1894. *Symbolic Logic*. 2nd ed. London: Macmillan.

Verburgt, L. M. 2022. *John Venn: A Life in Logic*. Chicago: University of Chicago Press.

Whittle, P. 2000. *Probability via Expectation*. 4th ed. New York: Springer. https://doi.org/10.1007/978-1-4612-0509-8.

Wilke, C. O. 2019. *Fundamentals of Data Visualization: A Primer on Making Informative and Compelling Figures*. Sebastopol, CA: O'Reilly.

Wilkinson, L. 2012. Exact and approximate area-proportional circular Venn and Euler diagrams. *IEEE Transactions on Visualization and Computer Graphics* 18: 321–331. https://doi.org/10.1109/TVCG.2011.56.

Wong, B. 2011. Points of view: Color blindness. *Nature Methods* 8: 441. https://doi.org/10.1038/nmeth.1618.

Youschkevitch, A. P. 1971. Euler, Leonhard. In Vol. 4 of *Dictionary of Scientific Biography*, ed. C. C. Gillispie, 467–484. New York: Charles Scribner's Sons.

12 Bibliographic note on Martin Gardner's columns

Martin Gardner's columns on "Mathematical Games" over many years in *Scientific American* covered much more than games and puzzles and included many splendid expositions of topics with mathematical content. They present a variety of small bibliographical challenges. The original articles will be accessible to many readers at https://www.jstor.org but typically under the titles "Mathematical Games". A further tiny detail is that pagination starts afresh in each issue of *Scientific American*, so volume and issue number together are needed for an exact citation. The columns were collected later in book form, often revised or retitled, in books that themselves often varied in publisher and even title over various reprints and reissues. A project to publish further revised editions, under yet other titles, from Cambridge University Press and the Mathematical Association of America, released its first four volumes between 2008 and 2014 but appears to have stalled. At the time of writing, it had not reached the books mentioned here.

https://en.wikipedia.org/wiki/List_of_Martin_Gardner_Mathematical_Games_ columns and https://ansible.uk/misc/mgardner.html will help you find what you are looking for or indeed to determine whether a relevant column was ever written. See Richards (2023) for a highly detailed bibliography of Gardner's publications.

About the authors

Nicholas J. Cox is a statistically minded geographer at Durham University. He contributes talks, postings, FAQs, and programs to the Stata user community. He has also coauthored 16 commands in official Stata. He was an author of several inserts in the *Stata Technical Bulletin* and is Editor-at-Large of the *Stata Journal*. His "Speaking Stata" articles on graphics from 2004 to 2013 have been collected as *Speaking Stata Graphics* (2014, College Station, TX: Stata Press), and he has edited *Stata Tips* (2 volumes, 2024, College Station, TX: Stata Press).

Tim P. Morris is a principal research fellow in medical statistics at University College London. He works on the development and evaluation of statistical methods for health research. His interests include estimands, missing data, simulation studies, covariate adjustment in randomized trials, meta-analysis, and the rerandomization design. He uses visualizations in all of these areas to facilitate explanation and understanding of ideas and results.