Product plots

Hadley Wickham, and Heike Hofmann

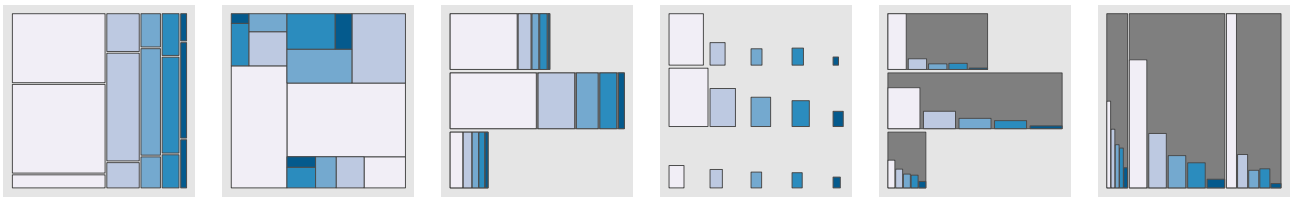

Fig. 1. A selection of plots encompassed by the product plots framework. All display the same data (the distribution of happiness and marital status), but each supports a different question. From left to right: mosaic plot, treemap, stacked bar chart, fluctuation diagram, and two new plots that don't have names. In all plots, area is proportional to probability.

Abstract—We propose a new framework for visualising tables of counts, proportions and probabilities. We call our framework product plots, alluding to the computation of area as a product of height and width, and the statistical concept of generating a joint distribution from the product of conditional and marginal distributions. The framework, with extensions, is sufficient to encompass over 20 visualisations previously described in fields of statistical graphics and infovis, including bar charts, mosaic plots, treemaps, equal area plots and fluctuation diagrams.

Index Terms—Statistics, joint distribution, conditional distribution, treemap, bar chart, mosaic plot

1 Introduction

Tables of counts, proportions and probabilities are an extremely common form of data, and many researchers have developed visualisations to display them. In this paper, we develop a framework that encompasses many existing visualisations, from bar charts to treemaps to pie charts, showing how graphics that previously seemed unrelated in fact share a deep underlying connection [1]. Our framework makes it easier to describe and create visualisations of categorical data, shows how existing methods are related and where new methods can be developed, and will make it easier to match questions about categorical data to the visualisations that will provide revealing answers.

Our framework focusses on area charts, where the area of a graphical element is proportional to the underlying count, proportion, or probability. We call our framework *product plots* in allusion to two products: the product of width and height to generate area, and the product of conditional and marginal distributions to produce joint distributions. A key development of the products plot framework is the inverse operation: the factorisation of high-dimensional data to products of low-dimensional plots. This allows us to combine simple, low-dimensional graphical primitives to display complex, high-dimensional data.

We begin in Section 2 with a review of related work. Then Section 3 motivates the three specific graphical constraints at the heart of product plots: the rectangle. We constrain the rectangle further to produce three 1d atoms (bar, spine, tile) and one 2d atom (the fluct). These constraints will later be relaxed in Section 7 to extend our framework to include more visualisations such as histograms, pie charts, cascaded treemaps and weighted plots.

Section 4 provides the mathematical framework that allows us to combine the 1d and 2d atoms to display data of any dimensionality. This framework is based on the fundamental statistical idea that any high dimensional distribution can be factorised into a product of low-dimensional conditional and marginal distributions. Section 5 shows how many existing named graphics are special cases of this general pattern.

Section 6 discusses some considerations for the display of product plots, and Section 8 introduces the R package `prodplot`, our reference implementation of the product plots framework. Finally, Section 9 discusses our plans for the future.

To illustrate these ideas, we will use the same data in all examples. The data is a small sample of variables related to happiness from the general social survey (GSS) [2]. The GSS is a yearly cross-sectional survey of Americans, run from 1976. We combine data for 25 years to yield 51,020 observations, and of the over 5,000 variables, we select nine related to happiness, as described in Table 1.

2 Related Work

The product plots framework is heavily influenced by the work of Wilkinson [4; 5], who suggested that both mosaic plots and treemaps can be described as plots with coordinate systems based on recursive partitioning. Our work builds on this by adding conditioning and defining a "grammar of area plots", a domain specific language that describes this small family of plots in more detail. Other similar efforts to build domain specific languages for visualisation are APT [6] and VisQL [7]: our framework is driven by similar forces, but carves out a much smaller niche. This makes products plots less expressive, but more efficient, because the number of primitives required to describe the smaller domain is much smaller.

The HiVE system [8] introduces a notation for describing the states of hierarchical layouts and operators for reconfiguration. Our work extends HiVE by adding conditioning, and including non-space filling visualisation. HiVE supports both rectangular layouts and polygon layouts. Of the rectangular layouts, all but the spatially ordered algorithm are included within the product plots framework.

Polaris [9] relies heavily on a tabular or cubic format in rendering hierarchical graphics, which makes these charts essentially a three dimensional extension of the trellis framework [10; 11]. In the product plots framework, this approach is captured by conditioning on the

- *Hadley Wickham is an Assistant Professor of Statistics at Rice University, Email: hadley@rice.edu.*
- *Heike Hofmann is an Associate Professor of Statistics at Iowa State University.*

Manuscript received 31 March 2011; accepted XX XXXX XXXX; posted online XX XXXX XXXX; mailed on XX XXXX XXXX.
For information on obtaining reprints of this article, please send email to: tvcg@computer.org.

Variable	Description	Values
age	age in years	18–89
degree	highest education	lt high school, high school, junior college, bachelor, graduate
finrela	relative financial status	far above, above average, average, below average, far below
happy	happiness	very happy, pretty happy, not too happy
health	health	excellent, good, fair, poor
marital	marital status	married, never married, divorced, widowed, separated
sex	sex	female, male
wtsall	probability weight	0.43–6.42
year	year of survey	1972–2006

Table 1. Description of sample data. Common plot colours are shown next to respective levels. Colours from ColorBrewer palettes [3].

variables used in rows and columns, and displaying the other variables unconditioned in each cell of the resulting tabular layout.

3 Graphical primitives

Area plots correspond to tilings of the (2d) plane. We could consider partitions of higher-d spaces (e.g. 3d or 4d), but given that we have to project these down to 2d for viewing on paper or screen, there is little disadvantage to working directly in 2d. There are many possible ways to tile the plane, so to cut these down to a manageable number, we identify constraints, in the style of [12], that are important for visualising counts.

Firstly, area must be **proportional** to count. This is the key constraint underlying all area plots. The total area for a graphic is usually constrained, which means that area plots display typically proportions or probabilities, rather than counts. With this proviso, we will use count, probability and proportion interchangeably throughout the paper.

Secondly, partitions must be **disjoint**. To be able to see the complete area, each rectangle must be non-overlapping. Note that this does not imply that the tiling must be space-filling, and two of the four primitives, described next, are not.

Finally, we require partitions to be **rectangular**. If partitions are rectangular, many interesting perceptual tasks only require comparing lengths, or positions along a common scale, tasks which are generally easier than comparing areas [13; 14]. There is little evidence to suggest that rectangles are "best" shape for comparison [15; 16], but they are computationally simple, recursive, in the sense that we can always tile a rectangle with smaller rectangles, and form the basis for many existing graphics.

These constraints give rise to four graphical primitives. Section 3.1 describes the three partitions of 1d data, bars, spines and tiles, and Section 3.2 describes the one partition of 2d data, the fluct. Each of the three constraints can also be relaxed, yielding the additional types of partitions described in Section 7.

3.1 1d primitives

1d primitives display 1d *data*, i.e. counts broken down by a single variable. There are three 1d primitives, as shown in Figure 2 and described below.

Fig. 2. 1d partitions showing the distribution of happiness. From left to right: bars, spines and tiles.

- **bar**s: height is proportional to value, width equally divides space. Bars are not space filling, occupying $\text{mean}(x - \max(x))$ of the total area. Bars can be arranged horizontally ("hbar") or vertically ("vbar").
- **spine**s: width is proportional to value, height occupies full range. Spines are space filling and can be arranged horizontally ("hspine"), vertically ("vspine"), or can automatically pick their orientation ("spine") by splitting the largest dimension. The name spine is evocative of books sitting on a library shelf [17]. Bars and spines are indistinguishable when the underlying data is evenly distributed across the categories.
- **tile**s: no restrictions on height or width, just tile the plane with rectangles, trying to keep the aspect ratio of each rectangle close to 1. This is partitioning defined by the squarified treemap [18].

Each of these three displays has different strengths and weaknesses. It is easiest to compare the value associated with bars because the perceptual task is the easiest: comparing position on a common scale. Reading spines and tiles are harder because we must compare lengths or areas, but they occupy the complete space and so work better recursively.

We can create 2d primitives by combining these 1d primitives, as shown in Section 4, but there is one 2d primitive that does not arise in this way.

3.2 2d primitives

2d primitives display 2d *data*, i.e. a count broken down by two variables. We are currently aware of only one primitive for 2d data, the **fluct**, derived from the fluctuation diagram [19]. The fluct has height and width proportional to the square root of the count. Each rectangle is arranged on a regular grid formed by the levels of the two variables, allowing comparisons both vertically and horizontally.

A special case of the fluct is the equal bin size plot [19] which occurs when the two variables are jointly uniformly distributed, usually as a result of the conditioning described in the following section. The equal bin size plot is particularly useful as a way of visualising missing combinations. Figure 3 illustrates these two types of graphics.

4 Probability and plot products

To construct plots of higher-dimensional data sets, we need a way to decompose them into 1d and 2d components. Some statistical vocabulary is useful: one way of describing the input data is as a **probability mass function**, or PMF. A PMF is a function with n inputs, each indexing one dimension with an integer, which outputs the probability of each combination of inputs. A PMF has two restrictions: every value must be greater than or equal to zero, and all values must sum to one.

4.1 Joint distributions are the product of marginals and conditionals

Figure 4 shows three ways to represent the the 2d table of proportions of sex and happiness. The top table displays the **joint distribution**, and allows us to answer questions of the form "what proportion of all people are male and very happy?" – the bottom left number tells us

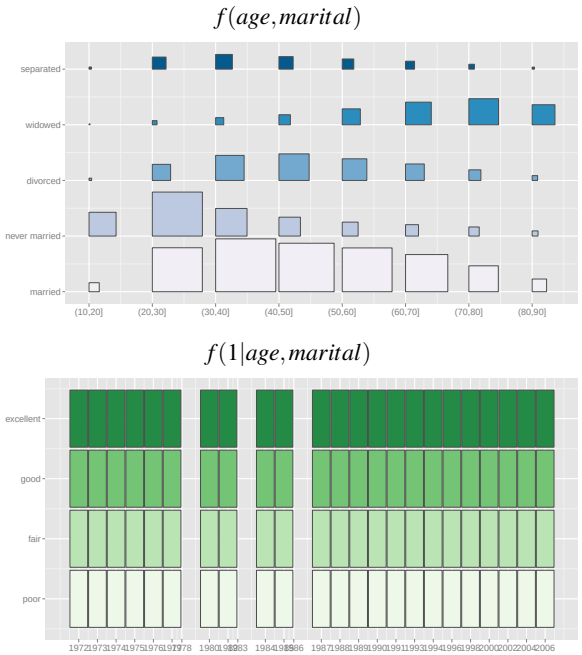

Fig. 3. (Top) A fluctuation diagram showing distribution of age (in decades) and marital status. (Bottom) Equal bin-size plot showing health status and survey year. Thee three empty columns show that health status was not recorded for three years.

that these are 0.14. The middle two tables displays two **conditional distributions**: the distribution of sex given happiness, and happiness given sex. These correspond to restricting the row or column sums to one, and support questions such as "what proportion of pretty happy people are female?" (the table tells us 0.55), or "what proportion of males are not too happy?" (the top left number in the table tells us that it's 0.12). Finally, the last table displays the two **marginal distributions** of sex and happiness. These allow us to answer questions like "what proportion of respondents were male" (0.44) or "what proportion of respondents were very happy?" (0.30).

Formally, a conditional distribution function is written $f(x|y)$ and is equal to $f(x,y)/f(y)$. This definition illustrates that an important statistical fact: given a conditional distribution and a marginal distribution, we can always find the joint distribution: $f(x,y) = f(x|y)\,f(y)$[1]. Conversely, we also see that we need to know both the marginal and conditional distributions in order to re-construct the joint distribution, i.e. only the joint distribution contains the full information about the relationship between all variables involved. Given the joint distribution $f(x,y)$, we can get either marginal distribution by integrating (or summing, in the case of a categorical variable) the other variable out: $f(x) = \int_y f(x,y)dy$. The marginal distribution together with the joint then lead to the conditional distribution. These definitions extend in a straightforward way to higher dimensions. For example, a 3d joint distribution can be written as the product of 2d and 1d conditional and marginal distributions in the following three ways:

- $f(x,y,z) = f(z)\,f(x,y|z)$
- $f(x,y,z) = f(y,z)\,f(x|y,z)$
- $f(x,y,z) = f(z)\,f(y|z)\,f(x|y,z)$

This means that we can build any high-dimensional PMF as a product of low-dimensional conditional and marginal PMFs, a computationally trivial operation. In the following, we are using the previously

[1]Because the parameters of a PMF identify it, statisticians often fail to explicitly label the different functions. To be precise, the above statement should be written: $f_{X,Y}(x,y) = f_{X|Y=y}(x|y)\,f_Y(y)$

$f(happy, sex)$

	male	female
not too happy	0.05	0.07
pretty happy	0.25	0.31
very happy	0.14	0.18

$f(sex|happy)$

	male	female
not too happy	0.43	0.57
pretty happy	0.45	0.55
very happy	0.43	0.57

$f(happy|sex)$

	male	female
not too happy	0.12	0.12
pretty happy	0.57	0.55
very happy	0.31	0.32

$f(happy)$ and $f(sex)$

	male	female	
not too happy			0.12
pretty happy			0.56
very happy			0.30
	0.44	0.56	

Fig. 4. The distribution of happiness and sex, displayed in three equivalent ways. (Top) Joint distribution. Overall table sums to one. (Middle) Conditional distribution of sex given happiness and marginal distribution of happiness. (Bottom) Conditional distribution of happiness given sex and marginal distribution of sex.

introduced low dimensional primitives for a graphical analog of this multiplication.

4.2 Area is the product of height and width

We connect probability products to our rectangular primitives by noting that areas are also products: products of height and width. It's easiest to show this with a picture: Figure 5 shows how our simple 1d primitives combine to get two familiar plots: the mosaic plot and the stacked bar chart. From left to right, we have plots of $f(happy)$, $f(sex|happy)$ and the product $f(happy, sex)$. The heights and widths of the rectangles are multiplied in the same way as the components of the PMF.

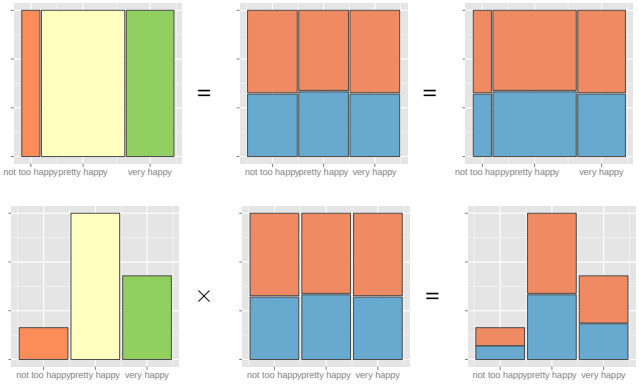

Fig. 5. Plots of the distribution of happiness and sex (■ male, ■ female) (Left) $f(happy)$, (Middle) $f(sex|happy)$, (Right) $f(happy, sex)$.

Figure 6 shows a more complicated example: visualising a 3d distribution as a product of three low-dimensional distributions. We first display $f(marital)$, then $f(marital, sex) = f(sex|marital)\,f(marital)$, and finally $f(marital, sex, happy) = f(happy|sex, marital)\,f(sex|marital)\,f(marital)$. This plot uses two vspines and an hspine to produce a mosaic plot.

More precisely, a product plot is constructed by the following recursive partitioning algorithm, which takes three parameters: data, a

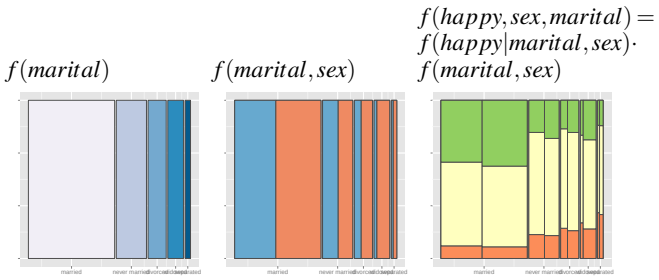

Fig. 6. Conditional on marital status, are men or women happier? This figure shows the construction of $f(happy, sex, marital)$ with (from left to right) a vspine by marital status, a vspine by marital status and sex, and a hspine by marital status, sex and happiness. For all levels of marital status, men are slightly less happy.

multi-dimensional array, with dimensions ordered in the same way as the desired partitions; bounds, a vector giving the top, left, right and bottom boundaries; and divider, a list of the desired low-d drawing primitives. The following algorithm gives an rough idea of the computation:

- Calculate the one marginal and i conditional distributions.
- For each probability in the marginal distribution, divide the current bounds into i sets of new bounds, one for each level of the categorical variable. The new bounds are found using the algorithm defined by the drawing primitive.
- For each new bound and matching conditional distribution, call the partition function recursively, dropping one element from the list of drawing primitives.

Different partitions reveal different features of the data. Take for example, the distribution of age and marital status, as shown in Figure 3. Instead of visualising the joint distribution with a fluct, we could focus on the conditional distribution of marital status given age, or age given marital status. Figure 7 shows two ways to do this. The left plot shows $f(marital, age)$ with a vspine nested in a hspine, and the right plots $f(age, marital)$ with a hbar nested in a vspine. These displays show the same data, but support different comparisons: on the left, we can see that most young people are unmarried, and on the right, that few unmarried people are over the age of 30. The right plot, with bars nested inside spines, also illustrates an important feature of non-space filling tilings: the relationship between proportion and area is only constant within a level. The bars are scaled to be as tall as possible, without overflowing any bounding region.

Conditioning is also an important tool by itself, because it allows one to remove relationships that are known or uninteresting. Figure 8 uses a fluct and a vspine to explore the relationship between happiness, health and financial status. The left plot displays raw proportions, showing that most people are in good health and average financial standing. However, it is difficult to see how happiness varies within these conditions because we must compare areas, not positions. Conditioning on financial status and health produces the plots on the right (equal bin size plots) and makes it easier to see the conditional distribution of happiness given sex and health, because comparing positions along a common scale is an easier perceptual task. Depending on the comparison we are most interested in, we can make it easier to compare across wealth given health, or health given wealth. Here we see that for a fixed income level, better health is correlated to increased happiness. The same is not true for a fixed level of health: rich people with poor health seem to be less happy than poorer people in poor health.

5 Existing plot types

Many existing plots fall into this framework. The low-d primitives already have their own names:

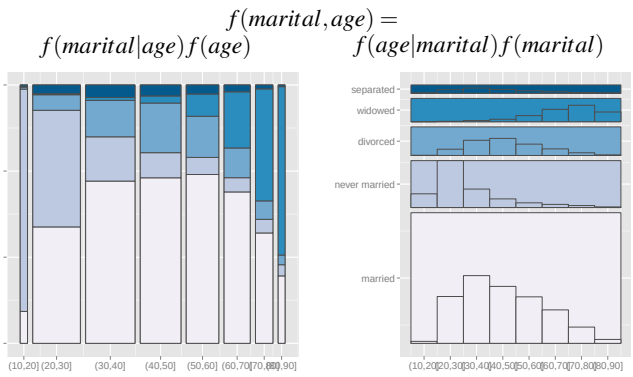

Fig. 7. (Left) The joint distribution of marital status and age, $f(marital, age)$ as $f(marital|age) \cdot f(age)$, (in decades) partitioned by a vspine and hspine. (Right) The joint distribution of age and marital status, $f(age, marital)$ as $f(ageT|marital) \cdot f(marital)$, partitioned by a vspine and hbar.

- **Bar chart** (1d). 1 hbar.
- **Column chart** (1d). 1 vbar.
- **Spineplot** (1d). 1 spine.
- **Fluctuation** diagram (2d): 1 fluct.

And many more plots correspond to higher order combinations:

- **Stacked** bar chart (2d). 1 hbar and 1 vspine.
- **Nested** bar chart [20] (2d). 2 hbars.
- **Equal bin size** plot [19] (3d): 1 fluct and 1 vspine, conditioned on the first two variables.
- **Mosaic** plot [21–23] (nd). Alternating hspines and vspines.
- **Double-decker** plot [24] (nd). $n-1$ hspines and 1 vspine.
- **Treemap** [25] (nd): n spines.
- **Squarified treemap** [18] (nd): n tiles.
- **Generalised treemaps** [26] (nd): any plot ending with a tile.

Trellis graphics [11], also known as latticed, facetted and conditioned graphics, are another related display. They use categorical variables to generate multiple panels, each containing a plot of the subset of the data. Trellised plots of area graphics also fall into our framework and can be created by conditioning on the trellising variables.

6 Display considerations

Labelling product plots is challenging. In this paper, we use a carefully selected combination of colour and axis labels, as well displaying the process by which a product plot is created. Axis labels are only available under certain conditions: when we have rows or columns that all share the same dimensional index. This can occur because of the structure of the graphical primitive (e.g. bars, flucts), or because the display is of a conditional distribution. Take Figure 7 for example: the left plot can only have the x axis labelled, but the right plot can have both axes labelled. There are other ways to label the cells apart from colour. [8] uses text labels, sized to occupy the space of the region they label. In the interactive setting, dynamic labelling in the form of tool tips is extremely helpful, as is the ability to use linked brushing to connect high-d plots to low-d plots.

We have observed that some aspect ratios are more aesthetically pleasing than others. This is particularly obvious for the fluctuation diagram where, in our experience, the plots are most appealing when the flucts are square. In other plots there does not seem to be an easy rule of thumb, but we wonder if aspect ratios close to the golden ratio might be more appealing. We are not aware of any previous work on aesthetics of aspect ratio, although they do seem to affect perception [27]. More research in this area is needed, particularly to explore the balance of aesthetics and usability.

Fig. 8. (Left) $f(happy, health, finrela) = f(happy|health, finrela) \times f(health, finrela)$, partitioned with a vspine and fluct. Health is on the x-axis, financial status on the y-axis. (Middle) $f(happy|health, finrela)$. We can no longer see the joint distribution of health and financial status, but it is much easier to see the conditional distribution of happiness. Healthier and richer people are happier: maybe money does buy happiness? (Right) $f(happy|health, finrela)$ partitioned with a fluct and **h**spine, emphasizing the relationship of happiness with finances, whereas the middle plot emphasizes the relationship with health. (■ Not too happy, ■ pretty happy, ■ very happy)

7 Variations and Extensions

On top of this basic framework, it is useful to consider a few variations and extensions. Instead of counts, we can plot weighted data or continuous data, or we can relax the display constraints to allow displays where area is not proportional to weight, partitions are non-disjoint or non-rectangular.

7.1 Weighting

We have assumed that the proportions represent counts, but without loss of generality, we can use any set of non-negative, additive weights. For example, in the happy dataset, the wtssall variable gives analytic survey weights. These are used to account for oversampling of black respondents in certain years, and to reduce the effect of non-response in other demographics. Figure 9 shows the difference between the weighted and unweighted distributions of age and sex. The distribution is barely different, and suggests that we don't need to worry about weights for this plot. (Unfortunately for this dataset of we have been unable to find any plots where weighting does makes a difference.) In other datasets, weights can be useful to move from numbers of counties to numbers of people, or to areas, or to other relevant quantities. Some examples of weighted data graphics can be found in [28–30].

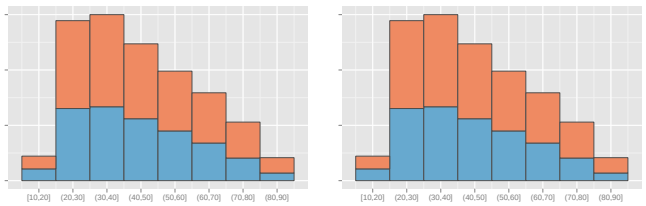

Fig. 9. Joint distribution of age and sex (■ male, ■ female), $f(age, sex)$. (Left) counts and (right) probability weights. In this case, there is very little difference between the plots.

7.2 Continuous data

The framework can be trivially extended to work with continuous data: just bin continuous variables to make them discrete. There are many different to ways to create bins for continuous data, but two are most important [31]: bins of equal width, and bins containing equal numbers of points. This extension allows the product plot framework to also describe **histograms** and **spinograms** [17], continuous analogues of bar and spine charts, shown in Figure 10. A long standing tradition is that no gaps are displayed between adjacent rectangles when displaying continuous data.

A more theoretical approach in dealing with continuous variables lies in increasing the number of bins infinitesimally, which leads from a probability mass function to a probability density function, turning the 1d primitives bar primitives to density plots, and the 1d spine primitives to a conditional density plot [32]. That approach lends itself to the inclusion of one continuous variable, but two variables need an additional aesthetic, such as colour, to visualize a 2d joint density. Three or more continuous variables break the current scope of the product plots framework.

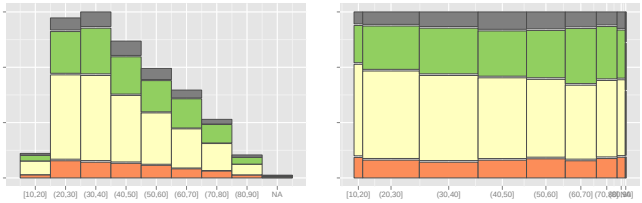

Fig. 10. The distribution of happiness with age, $f(age, happy)$. (Left) hbar + vspine: an extension of the histogram. (Right) hspine + vspine: an extension of the spinogram.

With one more extension, displaying the innermost proportion with colour (instead of area), we can also describe **dimensional stacking** [33].

7.3 Area not proportional to weight

It can be useful to violate the constraint that area is proportional to value to distinguish between zeros, missing values and very small values. A zero weight should have zero area, but giving it positive area can be useful so that it is visible. In general, it's useful to constrain all areas to be above a certain minimal perceptible size (e.g. 4 square pixels). Areas which are constrained in such a way need a visual flag (such as a different colour) to ensure that the reader knows that the

relationship between area and value has been violated. This type of non-linear mapping has been implemented in MANET [34] and in hierarchical pixel bar charts [12]. At the other end of the spectrum, it can be useful to constrain the size of largest values to get censored zooming [Antony Unwin, priv. comm.], which makes it possible to focus on small values.

Other non-linear transformations may also be useful. For example, we could take square roots to stabilise the variance of the areas. Tukey applied this technique to histograms to create rootograms [35; 36]: histograms where the y-axis has a square root scale.

7.4 Non-disjoint partitions

Cascaded treemaps [37] illustrate how the violation of containment can be productive. In the **cascaded treemap**, each level is slightly offset from the one above to create a pseudo-3d perspective. This makes it easier to see all the levels of the hierarchy, not just the lowest level. Figure 11 shows an example of how cascading can help illuminate the structure of a complex mosaic plot. This technique is probably most effective when implemented interactively.

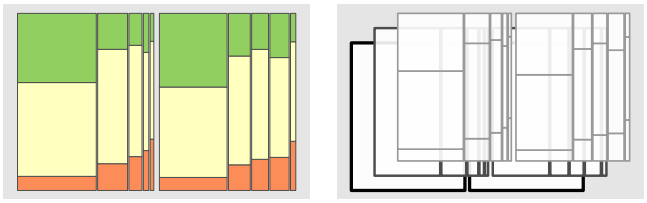

Fig. 11. A mosaic plot of happiness by marital status and sex, $f(happy, marital, sex)$. (Left) Coloured by happiness (■ Not too happy, ■ pretty happy, ■ very happy). (Right) A cascaded view helps show how the plot is built up.

7.5 Non-rectangular partitions

A pie chart is a popular method of displaying proportions, but it is not a rectangular partition and so does not seem to fall in the framework of this paper. However, there is a simple relationship between product plots and pie charts: a pie chart is an hspine drawn in polar coordinates with the x coordinate mapped to angle and the y coordinate to radius. Many other circular displays turn out to be special cases of product plots drawn in polar coordinates [38]. We have identified the following radial plots as polar transformations of product plots:

- **Wind rose** (aka sector graphic) [39] and **fourfold displays** [40] (2d): 1 hbar, and 1 vspine. Nightingales's coxcomb [41] is very similar, but the slices overlap and so violate the constraint of disjoint area.
- **Concentric pie chart** (aka bullseye chart) (1d): 1 hspine.
- **Doughnut plot** (2d): 1 hspine, and 1 vspine.
- **Racetrack plot** (aka circular bar chart) (1d): 1 vbar.
- **Infoslices** [42] (nd): n vbars. But they only use half of the polar plane, and are specialised for highly nested data.

Eight polar variants are displayed in Figure 12. Many of these are familiar and already have names. They are all of dubious utility because research suggests that visualisations in polar coordinates are harder to reader accurately than visualisations in Cartesian coordinates [43].

Generally, the y axis (mapped to radius) must be square-root transformed to ensure that that counts stay proportional to areas. Fan-lenses [44] and Stasko's radial displays [45] deliberately do not do this in order to emphasise the outer levels.

Other non-rectangular area graphics include non-rectangular treemaps, such as circular treemaps [46], space-filling curves [47] and voronoi treemaps [48].

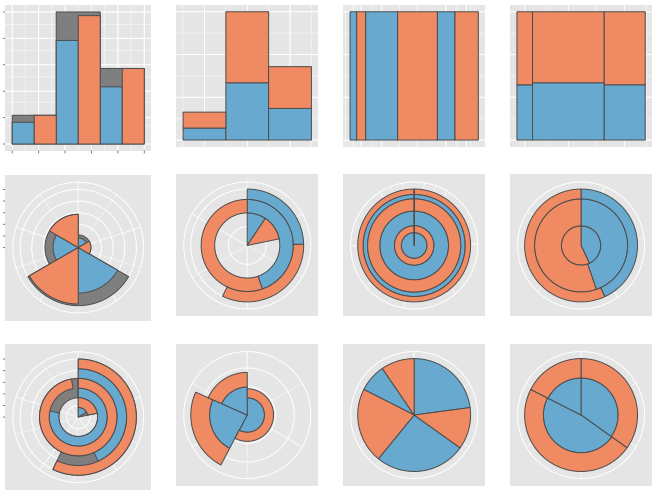

Fig. 12. (Top) Area graphics in Cartesian coordinates. (Mid) Area graphics in polar coordinates. (Bottom) Product graphics with alternate mapping of x and y to r and θ. From left to right: hbar + hbar, vspline + hbar, hspine + hspine, hspine + vspine. (■ male, ■ female)

8 R PACKAGE

We have provided a reference implementation of these ideas in an R [49] package called `productplots`, available from `http://github.com/hadley/productplots`. There are two main functions: `prodcalc`, which computes the coordinates of each rectangle; and `prodplot`, which displays the rectangles with the ggplot2 package [50]. The code is well tested, and ensures that the constraints are always satisfied.

For example, the following code conveys the essence of the plots in Figure 8. The function `prodplot` creates plots of the `happy` data, defined using the standard R formula notation (`~`), with the convention that `|` denotes conditioning. The final argument lists the graphical primitives to use for display. This can also be a template function like `mosaic()` which produces a standard, named, graphic.

```
prodplot(happy, ~ happy + finrela + health,
  c("vspine", "fluct"))
prodplot(happy, ~ happy | finrela + health,
  c("vspine", "fluct"))
prodplot(happy, ~ happy | finrela + health,
  c("hspine", "fluct"))
```

As for all presentation graphics, the actual code is somewhat more complicated, as we make a number of tweaks for optimal display. The complete `productplots` code to used to create the images in this paper is available from the authors upon request.

The `productplots` package has been designed to be flexible and extensive. For example, each graphical primitive is represented by a function: `hspine()`, `vspine()`, `spine()`, `hbar()`, `vbar()`, `tile()` and `fluct()`. Adding a new graphical primitive is easy: you just write a new function, and can use the existing calculation and display algorithms.

9 CONCLUSION

The product plots framework is successful at describing many existing graphics that display tables of counts, proportions and probabilities. It lays the framework for much future work, particularly because the framework leads to a combinatorial explosion of possibilities. For example, a 4d PMF, $f(a,b,c,d)$, can be factorised in five different ways:

- $f(a,b,c,d) = f(a|b,c,d)f(b|c,d)f(c|d)f(d)$
- $f(a,b,c,d) = f(a|b,c,d)f(b|c,d)f(c,d)$

- $f(a,b,c,d) = f(a|b,c,d)f(b,c|d)f(d)$
- $f(a,b,c,d) = f(a,b|c,d)f(c|d)f(d)$
- $f(a,b,c,d) = f(a,b|c,d)f(c,d)$

There are 24 possible ways of ordering the variables in the PMF, 5 ways of displaying a 1d PMF, and 1 way of displaying a 2d PMF, leading to a possible $24 * (5^4 + 5^2 + 5^2 + 5^2 + 1) = 16,824$ plots, before we even consider conditioning! The product plots framework defines a large space of potential plots.

Well thought out interaction will make it easier to navigate this space, and we have begun to develop a prototype model in R, as part of the cranvas suite of interactive graphics, `http://github.com/ggobi/cranvas`. However, much work and user testing remains to be done before we can be confident that we have developed a useful navigation model.

We are also exploring the connection between product plots and log-linear (aka Poisson) models, the statistical models most commonly used for count data. Some special cases of the general connection have already been worked out. For example, [23] shows how looking for straight lines in a mosaic plot corresponds to a formal test of independence between two variables. Generally, we are interested in knowing what visual comparisons are equivalent to what formal statistical tests, and conversely, how significant coefficients in a model can help us choose a set of useful plots. Given a question, can we suggest appropriate plots? Given a plot, can we suggest questions that it might answer?

Product plots also need to be extended with systematic ways of displaying uncertainty, to help users identify whether differences in the plot represent real differences in the underlying population or are just the result of random variation. This is particularly important because area plots typically have fixed area, meaning that the total number of observations in a plot is not directly displayed, even though this is crucial for determining whether a difference is statistically significant or not.

Finally, continuous data is currently incorporated in a simplistic manner. Can we extend the product plots framework to include plots designed to show continuous distributions like the average shifted histogram [51] or the kernel density estimate [52]?

ACKNOWLEDGMENTS

Computations performed in R [49] and graphics produced with ggplot2 [50].

REFERENCES

[1] D. R. Cox, "Some remarks on the role in statistics of graphical methods," *Applied Statistics*, vol. 27, no. 1, pp. 4–9, 1978.

[2] J. A. Davis and T. W. Smith, "General social surveys, 1972-2008 [machine-readable data file]," 2008. Principal Investigator, James A. Davis; Director and Co-Principal Investigator, Tom W. Smith; Co-Principal Investigator, Peter V. Marsden; Sponsored by National Science Foundation. –NORC ed.– Chicago: National Opinion Research Center [producer]; Storrs, CT: The Roper Center for Public Opinion Research, University of Connecticut [distributor], 2009.

[3] C. A. Brewer, "Color use guidelines for mapping and visualization," in *Visualization in Modern Cartography* (A. MacEachren and D. Taylor, eds.), pp. 123–147, Elsevier Science, 1994.

[4] L. Wilkinson, *The Grammar of Graphics*. Statistics and Computing, Springer, 1999.

[5] L. Wilkinson, A. Anand, and R. Grossman, "Graph-theoretic scagnostics," in *IEEE Symposium on Information Visualization*, pp. 157–164, 2005.

[6] J. D. Mackinlay, "Automating the design of graphical presentations of relational information.," *ACM Transactions on Graphics*, vol. 5, no. 5, pp. 110–141, 1986.

[7] J. Mackinlay, P. Hanrahan, and C. Stolte, "Show me: Automatic presentation for visual analysis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1137–1144, 2007.

[8] A. Slingsby, J. Dykes, and J. Wood, "Configuring hierarchical layouts to address research questions," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, 2009.

[9] C. Stolte, D. Tang, and P. Hanrahan, "Multiscale visualization using data cubes," *Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on*, pp. 7 –14, 2002.

[10] W. Cleveland, *The Elements of Graphing Data*. Hobart Press, 1985.

[11] R. A. Becker, W. S. Cleveland, and M.-J. Shyu, "The visual design and control of trellis display," *Journal of Computational and Graphical Statistics*, vol. 5, no. 2, pp. 123–155, 1996.

[12] D. A. Keim, M. C. Hao, and U. Dayal, "Hierarchical pixel bar charts," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 8, no. 3, pp. 255 –269, 2002.

[13] W. S. Cleveland and R. McGill, "Graphical perception: Theory, experimentation and application to the development of graphical methods.," *Journal of the American Statistical Association*, vol. 79, no. 387, pp. 531–554, 1984.

[14] W. S. Cleveland and R. McGill, "An experiment in graphical perception," *International Journal of Man-Machine Studies*, vol. 25, no. 5, pp. 491 – 500, 1986.

[15] J. Hollands and I. Spence, "Judging proportion with graphs: The summation model," *Applied Cognitive Psychology*, vol. 12, no. 2, pp. 173–190, 1998.

[16] D. Simkin and R. Hastie, "An information-processing analysis of graph perception," *Journal of the American Statistical Association*, vol. 82, no. 398, pp. 454–465, 1987.

[17] J. Hummel, "Linked bar charts: Analysing categorical data graphically.," *Journal of Computational Statistics*, vol. 11, pp. 23–33, 1996.

[18] M. Bruls, K. Huizing, and J. J. van Wijk, "Squarified treemaps," in *IEEE Symposium on Information Visualization*, 1999.

[19] H. Hofmann, "Exploring categorical data: Interactive mosaic plots," *Metrika*, vol. 51, no. 1, pp. 11–26, 2000.

[20] J. Peltier, "Marimekko replacement – overlapping bars (easy)," 2009. URL http://peltiertech.com/WordPress/marimekko-replacement-overlapping.

[21] J. A. Hartigan and B. Kleiner, "Mosaics for contingency tables," in *Computer Science and Statistics: Proceedings of the 13th Symposium on the Interface*, (Fairfax Station, VA), pp. 268–273, Interface Foundation of North America, Inc., 1981.

[22] M. Friendly, "Mosaic displays for multi-way contingency tables," *Journal of the American Statistical Association*, vol. 89, no. 425, pp. 190–200, 1994.

[23] H. Hofmann, "Constructing and reading mosaicplots," *Computational Statistics and Data Analysis*, vol. 43, no. 4, pp. 565–580, 2003.

[24] H. Hofmann, "Generalized odds ratios for visual modeling," *Journal of Computational and Graphical Statistics*, vol. 10, no. 4, pp. 628–640, 2001.

[25] B. Shneiderman, "Tree visualization with tree-maps: 2-d space-filling approach," *ACM Trans. Graph.*, vol. 11, no. 1, pp. 92–99, 1992.

[26] R. Vliegen, J. J. van Wikj, and E.-J. van der Linden, "Visualizing business data with generalized treemaps," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 789–796, 2006.

[27] N. Kong, J. Heer, and M. Agrawala, "Perceptual Guidelines for Creating Rectangular Treemaps," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 16, no. 6, pp. 990–998, 2010.

[28] A. Unwin and H. Hofmann, "New interactive graphics tools for exploratory analysis of spatial data," in *Innovations in GIS 5*, pp. 46–55, London: Taylor Francis, 1998.

[29] A. R. Unwin, C. Volinsky, and S. Winkler, "Parallel coordinates for exploratory modelling analysis," *Computational Statistics & Data Analysis*, vol. 43, no. 4, pp. 553–564, 2003.

[30] A. R. Unwin, M. Theus, and H. Hofmann, *Graphics of Large Datasets*. Springer, 2006.

[31] L. Denby and C. Mallows, "Variations on the histogram," *Journal of Computational and Graphical Statistics*, vol. 18, no. 1, pp. 21–31, 2009.

[32] H. Hofmann and M. Theus, "Interactive graphics for visualizing conditional distributions," *Unpublished manuscript*, 2005.

[33] J. LeBlanc, M. Ward, and N. Wittels, "Exploring n-dimensional databases," in *Proceedings of Visualization '90*, pp. 230–237, 1990.

[34] A. R. Unwin, G. Hawkins, H. Hofmann, and B. Siegl, "Interactive graphics for data sets with missing values - MANET," *Journal of Computational and Graphical Statistics*, vol. 5, no. 2, pp. 113–122, 1996.

[35] H. Wainer, "The suspended rootogram and other visual displays: An empirical validation," *The American Statistician*, vol. 28, no. 4, pp. 143–145, 1974.

[36] J. W. Tukey, *Exploratory Data Analysis*. Addison–Wesley, preliminary ed., 1971.

[37] H. Lü and J. Fogarty, "Cascaded treemaps: examining the visibility and stability of structure in treemaps," in *Proceedings of Graphics Interface 2008*, pp. 259–266, Canadian Information Processing Society, 2008.

[38] G. Draper, Y. Livnat, and R. Riesenfeld, "A survey of radial methods for information visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 5, pp. 759 –776, 2009.

[39] L. Lalanne, *Appendice Sur La Representation Graphique Des Tableaux Météorologiques Et Des Lois Naturelles En Général*. Cours Complet de Météorologie, 1843.

[40] M. Friendly, "A fourfold display for 2 by 2 by k table," Tech. Rep. 217, Psychology Department, York University, 1995.

[41] F. Nightingale, *Notes on matters affecting the health, efficiency and hospital administration of the British Army*. London: Private publication, 1857.

[42] K. Andrews and H. Heidegger, "Information slices: Visualising and exploring large hierarchies using cascading, semi-circular discs," in *Proc. IEEE InfoVis*, 1998.

[43] S. Diehl, F. Beck, and M. Burch, "Uncovering Strengths and Weaknesses of Radial Visualizations—an Empirical Approach," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 16, no. 6, pp. 935–942, 2010.

[44] X. Lou, S. Liu, and T. Wang, "Fanlens: Dynamic hierarchical exploration of tabular data," in *Infovis 2007 (poster)*, 2007.

[45] J. Stasko and E. Zhang, "Focus+context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations," in *Proc. IEEE InfoVis*, 2000.

[46] K. Wetzel, "pebbles - using circular treemaps to visualize disk usage," 2008. URL http://lip.sourceforge.net/ctreemap.html.

[47] M. Wattenberg, "A note on space-filling visualizations and space-filling curves," in *Proc. IEEE InfoVis*, 2005.

[48] M. Balzer and O. Deussen, "Voronoi treemaps," in *Proc. IEEE InfoVis*, 2005.

[49] R Development Core Team, *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2010. ISBN 3-900051-07-0.

[50] H. Wickham, *ggplot2: Elegant graphics for data analysis*. useR, Springer, July 2009.

[51] D. W. Scott, "Averaged shifted histograms: Effective nonparametric density estimators in several dimensions," *The Annals of Statistics*, vol. 13, pp. 1024–1040, 1985.

[52] D. W. Scott, *Multivariate density estimation*. Wiley Online Library, 1992.