Colloquium Biometricum 40
2010, 31–40

VISUALIZING BIVARIATE RELATIONSHIPS WITH HEXAGONALLY BINNED DATA

Marcin Kozak[1], Agnieszka Wnuk[1], Dariusz Gozdowski[1], Zdzisław Wyszyński[2]

[1]Department of Experimental Design and Bioinformatics
[2] Department of Agronomy
Warsaw University of Life Sciences
Nowoursynowska 159, 02-776 Warsaw
e-mail: nyggus@gmail.com

Summary

Scatterplots are overwhelmingly often used in numerous branches of science. Unfortunately, they fail when the number of points in a plot is very large. Graphing hexagonally binned data is one of the efficient ways of dealing with such situations. This paper shows the application of a display of hexagonally binned data for a three-year field experiment with spring barley. The display is compared with regular scatterplots and it is shown that it provides information about a relationship when regular scatterplots fail, even despite of employing very small symbols and jittering. We also provide the code that was used to produce the graphs in R environment.

Key words and phrases: scatterplot, graphics, visualization

Classification AMS 2010: 62–09

1. Introduction

Scatterplots are commonly used to visualize bivariate or multivariate (by means of scatterplot matrices) data. Their advantages cannot go unnoticed: they are one of the best tools to show patterns and regularities in a bivariate data set (Cleveland 1994, Jacoby 1997), so their overwhelming applications in both exploratory data analysis and data reporting cannot surprise.

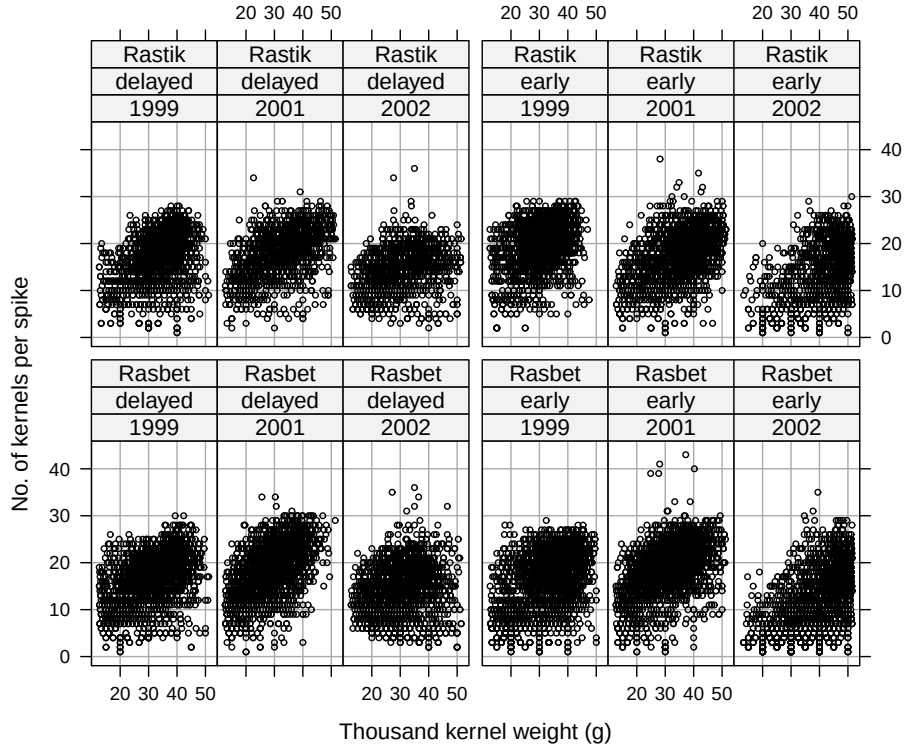

Fig. 1. A trellis display of scatterplots of number of kernels per spike versus mean thousand kernel weight for combinations of cultivar, sowing date and year; for each such combinations, data for four N doses (0, 30, 60 and 90) are pooled

Scatterplots do have, however, an important drawback when there are very many points to be presented in one plot: relationships between the variables are difficult to grasp (e.g., Martinez and Martinez 2005, Davis et al. 2006). See for example Figure 1, picturing a relationship between number of kernels per spike and thousand kernel weight in combinations of cultivar (Rasbet and Rastik), sowing date (early and delayed) and year (1999, 2001 and 2002). Refer to Gozdowski et al. (2007) for a detailed description of this split-plot experiment, in which the cultivar × sowing date combination constituted main plots, whereas nitrogen rates (0, 30, 60 and 90 kg/ha) the subplots; there were four replications for each combination. At harvest, for each replication plants were taken for measurements from two 1-meter length rows (giving the area of 0.22 m^2); and among other traits, number of kernels per spike and thousand kernel weight were observed for each spike (in this paper we use only these two traits). For the present paper, we pooled the data from the N doses so that in each such combination there

were observations for many spikes (ranging from 1163, for Rastik, early sowing date in 2002, up to 2479, for Rasbet, early sowing data in 2001).

From Figure 1 it is difficult to say anything about the relationships. Some of them look slightly linear (e.g., for Rasbet and Rastik, delayed sowing date in 2001), but one cannot be sure of that because of the serious overlap of the points. We can try to reduce the size of the plotting symbols, but that does not seem to help here, still due to the overlap of points—see Figure 2. Sometimes jittering (Cleveland 1994) can help—see Figure 3. Here it helped a little, but still the relationships are hidden behind all the points and their overlap; it is rather difficult to grasp the relationships here.

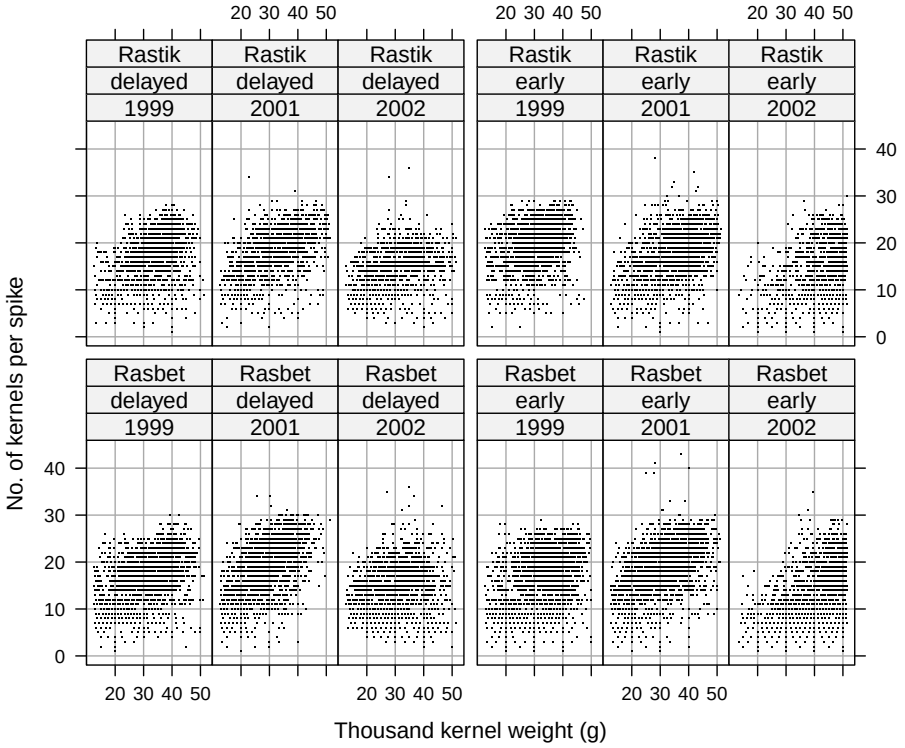

Fig. 2. The same scatterplots as in Figure 1, but with a dot (·) representing points

An efficient way of visualizing such data is by plotting binned data instead of all the points (Carr et al. 1987); unfortunately, this graphical method is not too popular, and quite likely is little known among researchers. The aim of this paper, then, is to present the application of a display of hexagonally binned data (Carr et al. 1987) for a three-year field experiment with spring barley, and com-

pare its usefulness with a regular scatterplot. This will be done in the second section; in the third section we will provide R (R Development Core Team 2009) code for the graphs, while in the fourth we will summarize the results.

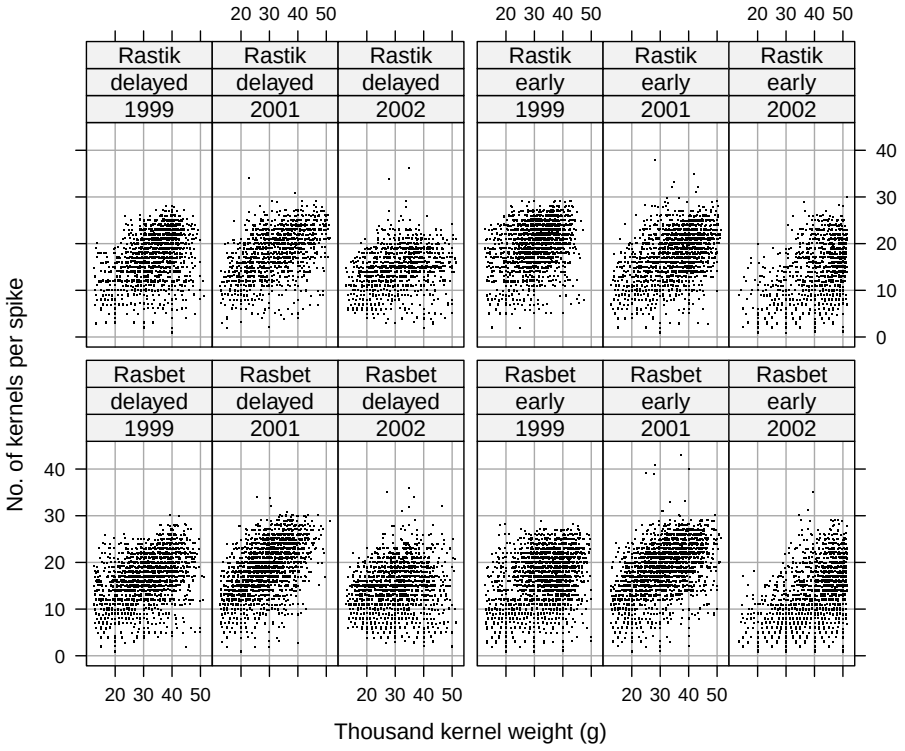

Fig. 3. The same scatterplots as in Figure 2, but with jitter added to no. of kernels per spike in order to reduce the effect of the overlap of the points

2. Hexagonal binning

The idea behind graphing binned data is very simple: instead of plotting single points, they are combined into bins, and the bins are plotted with symbols that are either sized or colored according to the frequency within them. Despite this simplicity, the technique is very efficient. Unfortunately, it has rarely been used in life sciences. For example, Genton et al. (2006) applied it for the analysis of spread of fires, with longitude latitude forming x- and y-axes, respectively; the frequency in bins was represented by the bins' size. Kraja et al.

(2007) used a similar method for the scatterplot of various inflammation and procoagulation biomarkers against the National Cholesterol Education Program (NCEP) metabolic syndrome categories, which is a qualitative variable. Another type of hexagonal binning is the density distribution sunflower plot, also used as an alternative to scatterplots. Dupont and Plummer (2003) applied it to picture a relationship between BMI and diastolic blood pressure for subjects in the Framingham Heart Study (Anonymous 1997); the researchers used both color and symbols to determine the frequency within the bins.

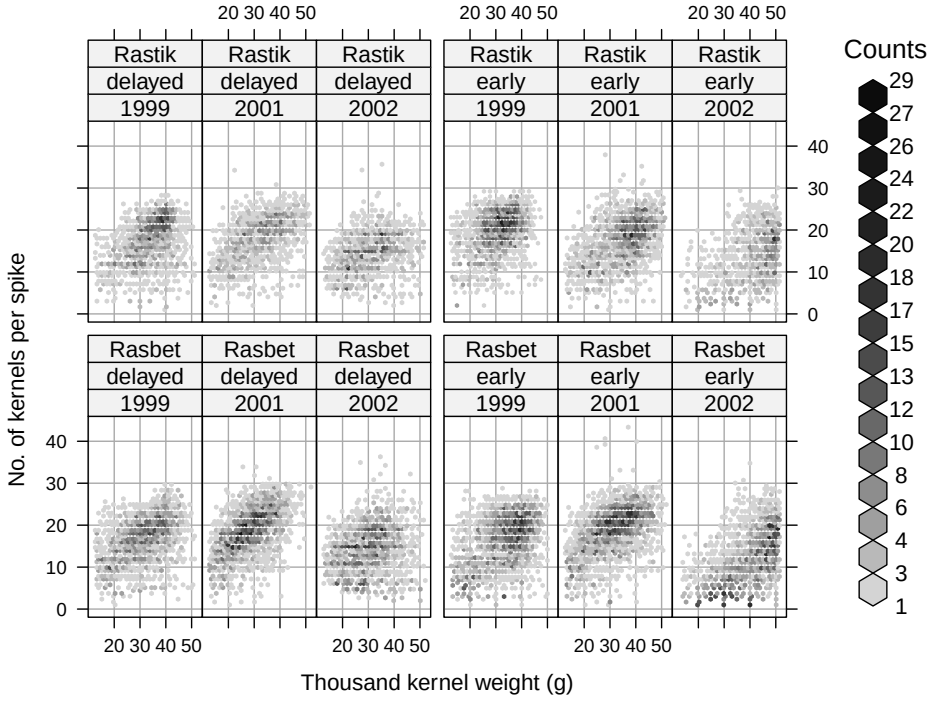

Fig. 4. A trellis display of plots of hexagonally binned data for the same data as plotted in Fig. 1

As suggested by Carr et al. (1987) and discussed by Wilkinson (2005), hexagonal bins are more efficient than rectangular ones in helping the viewer seeing the patterns in data. Here we will follow this advice and use hexagonal binning, with shades of grey representing bin frequency. The algorithm used for hexagon binning is reported in detail by Lewin-Koh (2009), distributed as a vignette with the hexbin package (Carr et al. 2009) of R. Figure 4 shows how this technique works. As shown by legend, darker bins represent higher density.

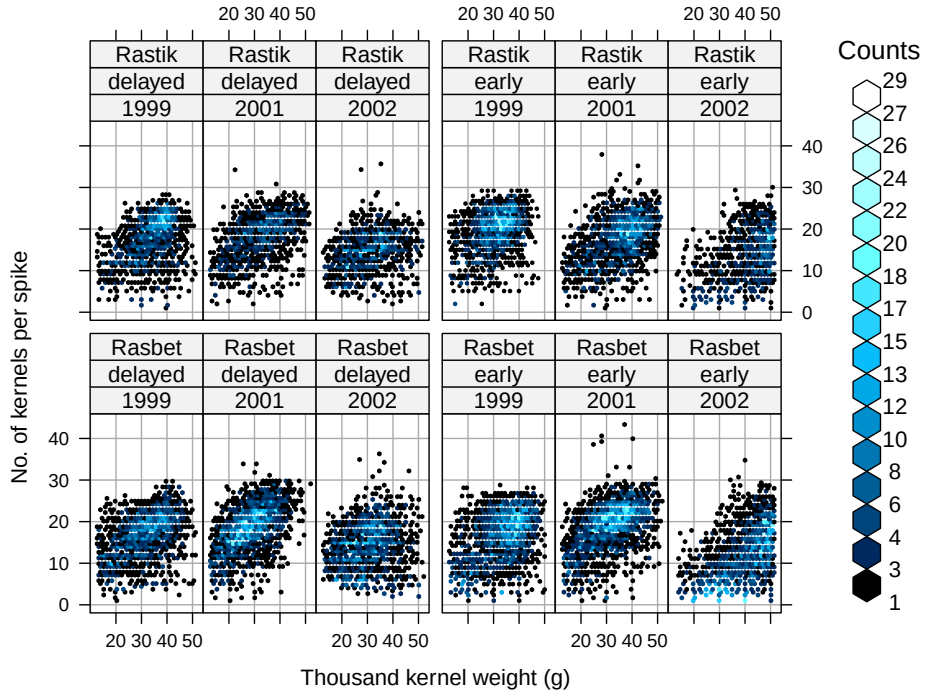

Fig. 5. The same display as that in Fig. 4, but with a blue to cyan colormap (the online version of the papers is in color)

Figure 5 is very similar, but instead of shades of grey, a blue to cyan colormap is used, with the brightest color representing the highest density.

Clearly the plots in Figures 4 and 5 represent the data better, showing regions of higher and lower density, which information was rather difficult to notice in Figures 1-3. Compare, for example, panels for Rastik and Rasbet, early sowing data in 2002 (two right panels in the plots). In Figures 1-3 these two panels do not seem to differ, but from Figures 4 and 5 it is clear that this is not true—for Rasbet for much more spikes a very small number of kernels was observed, which was the case for rather a small number of spikes for Rastik. Now let us compare panels for Rasbet, delayed sowing date in 2001, and Rastik, early sowing date in 2001. Although in Figures 1-3 they look quite similar, Figures 4 and 5 reveal that for the latter for more spikes greater thousand kernel weight was observed than was the case for the latter. In summary, from the figures we see that number of kernels per spike and thousand kernel weight are usually positively related, although the relationship varies from very week (e.g.,

for Rastik, early sowing date in 1999) to quite strong (e.g., for Rasbet, delayed sowing date in 2001); some specific relationships were also noticed (e.g., for both cultivars in 2002 for early sowing date).

3. Software and code

Figures 1-3 were constructed with the function xyplot of the lattice package (Sarkar 2008) of R (R Development Core Team 2009), while Figures 4 and 5 with the hexbinplot function of the hexbin package (Carr et al. 2009) of R. Here is the code we used to produce the figures.

First, let us see the structure of the data.bin data frame (note that it was prepared before the analysis in order to make the year a factor instead of a numeric variable):

```
> str(data.bin)
'data.frame'    : 22573 obs. of  5 variables:
 $ year         : Factor w/ 3 levels "1999","2001",..: 1 1 1 1
1 1 1 1 1 1 ...
 $ cultivar     : Factor w/ 2 levels "Rasbet","Rastik": 2 2 2
2 2 2 2 2 2 ...
 $ sowing.date  : Factor w/ 2 levels "delayed","early": 2 2 2
2 2 2 2 2 2 ...
 $ kernels      : int  5 5 7 12 7 12 13 14 17 12 ...
 $ kernel.weight : num  0.014 0.014 0.0171 0.0192 0.02 ...
```

We will need `lattice` and `hexbin` packages:

```
> library(lattice)
> library(hexbin)
```

In case of lack of the hexbin package (lattice is supplied with the basic installation of R), one can simply install it by typing

```
> install.packages("hexbin")
```

Now let us construct Figure 1, which will later be slightly updated to construct Figures 2 and 3:

```
> figure1 <-
 xyplot(kernels ~ 1000*kernel.weight | year + sowing.date +
cultivar,
 data = data.bin,
 type = c("p", "g"), layout = c(6, 2), col = 1,
 ylab = "No. of kernels per spike", cex = 0.5,
 xlab = "Thousand kernel weight (g)",
 between = list(x = c(0, 0, 0.5, 0, 0), y = 0.5))
```

We can now print the figure within the console:

```
> print(figure1)
```

or to pdf:

```
> trellis.device("pdf", file = "Figure 1.pdf", width = 7,
height = 6)
> print(figure1)
> dev.off()
```

Now let us draw Figure 2; for this, we only need to update the `figure1` object:

```
> figure2 <- update(figure1, pch = ".")
> print(figure2) # or to pdf as before
```

Figure 3 is produced by updating Figure 2 by adding a small amount of random noise to number of kernels per spike, so-called jitter (Cleveland 1994); however, the function `trellis.update`, used before, can change neither the formula nor the data, so we have to produce the plot from scratch:

```
> figure3 <-
   xyplot(jitter(kernels) ~ 1000*kernel.weight | year + sow-
   ing.date + cultivar,
   data = data.bin,
   type = c("p", "g"), layout = c(6, 2), col = 1,
   ylab = "No. of kernels per spike", pch = ".",
   xlab = "Thousand kernel weight (g)",
   between = list(x = c(0, 0, 0.5, 0, 0), y = 0.5))
> print(figure3)
```

Now let us draw Figure 4, with the help of `hexbin` packake. Fortunately, it is based on `lattice`, and so the formulas will not differ too much from those above:

```
> figure4 <-
   hexbinplot(kernels ~ 1000*kernel.weight | year + sow-
   ing.date + cultivar,
   data = data.bin,
   layout = c(6,2), type = "g",
   ylab = "No. of kernels per spike",
   xlab = "Thousand kernel weight (g)",
   between=list(x = c(0,0,0.5,0,0), y = 0.5))
> print(figure4)
```

We can finally use color (here we use the black to cyan colormap) instead of shades of grey, and add the border to the bins in the legend:

```
> figure5<- update(figure4, colramp = BTC)
> figure5$legend$right$args$colramp <- BTC
> fig5$legend$right$border <- "black"
> print(figure5)
```

And this is it. If someone wants to get rid of the grid from the panels (which actually *is* very helpful—see Cleveland 1994 for the discussion), then it suffices to update `figure1-figure3` objects by

```
> update(figure1, type = "p")
```

For `figure4` (or `figure5`), we need to do the following:

```
> update(figure4, type = NULL)
```

Or alternatively, one can simply remove the `type` argument from the original functions whatsoever.

4. Conclusion

Hexagonal binning can be very helpful in discovering patterns from large data sets. Of course, it is not the only type of plot that can be employed, one of examples being a plot of bivariate kernel density estimate. However, the display of hexagonally binned data has this important advantage that it resembles a scatterplot, so its interpretation is rather easy (Wilkinson 2005). Although practically nothing interesting could be seen from scatterplots in Figures 1-3, hexagonally binned data plotted in Figures 4 and 5 proved to be efficient in showing the relationships. Constructing graphs in R is fairly straightforward once one knows a general formula framework for the `lattice` package, which itself is quite straightforward too.

References

Carr D., Littlefield R.J., Nicholson W.L., Littlefield J.S. (1987). Scatterplot matrix techniques for large *N*. *Journal of the American Statistical Association* 82(398), 424–436.

Carr D., ported by Lewin-Koh N. and Maechler M. (2009). hexbin: Hexagonal Binning Routines. R package version 1.20.0. http://CRAN.R-project.org/package=hexbin.

Cleveland W.S. (1994). *The Elements of Graphing Data*. 2nd ed. Summit, NJ: Hobart, USA.

Davis J.Ch., Paul A., Ferl R.J., Meisel M.W. (2006). Topographical imaging technique for qualitative analysis of microarray data. *BioTechniques* 41, 554–558.

Dupont W.D., Plummer W.D. Jr. (2003). Density Distribution Sunflower Plots. *Journal of Statistical Software* 8 (3), 1–5.

Anonymous (1997). The Framingham Study – 40 Year Public Use Data Set, Bethesda, MD: National Heart, Lung, and Blood Institute, NIH.

Genton M.G., Butry D.T., Gumpertz M.L., Prestemon J.P. (2006). Spatio-temporal analysis of wildfire ignitions in the St Johns River Water Management District, Florida. *International Journal of Wildland Fire* 15, 87–97.

Gozdowski D., Kozak M., Kang MS., Wyszyński Z. (2007). Dependence of grain weight of spring barley genotypes on traits of individual stems. *Journal of Crop Improvement* 21 (1/2), 223–233.

Jacoby W. (1997) *Statistical Graphics for Univariate and Bivariate Data*. Sage University Papers Series. No. 07–117.

Kraja A.T., Province M.A., Arnett D., Wagenknecht L., Tang W., Hopkins P.N., Djoussé L., Borecki I.B. (2007). Do inflammation and procoagulation biomarkers contribute to the metabolic syndrome cluster? *Nutrition & Metabolism* 4:28, 1–12.

Lewin-Koh N. (2009). *Hexagon Binning: An Overview*. Technical Report [http://www.bioconductor.org/packages/2.4/bioc/html/hexbin.html], Nov 12, 2009.

Martinez W.L., Martinez A.R. (2005). *Exploratory data analysis with MATLAB*. Chapman &. Hall/CRC Press.

R Development Core Team (2009). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.

Sarkar D. (2008). Lattice. *Multivariate Data Visualization with R*. Springer.

Wilkinson L. (2005). *The Grammar of Graphics*. 2nd ed. Springer.

WIZUALIZACJA DWUWYMIAROWYCH ZALEŻNOŚCI PRZY POMOCY DANYCH POGRUPOWANYCH SZEŚCIOKĄTNIE

Streszczenie

Wykresy rozrzutu są niezwykle często wykorzystywane w wielu dziedzinach nauki. Niestety kiedy na wykresie znajduje się bardzo dużo punktów, ich użyteczność jest niewielka. W takich sytuacjach użyteczne może okazać się tworzenie wykresu dla danych pogrupowanych w sześciokąty. W tym artykule przedstawiamy zastosowanie takiego wykresu dla trzyletniego doświadczenia polowego z jęczmieniem jarym. Porównanie wykresu z klasycznym wykresem rozrzutu wykazało, że dane pogrupowane w sześciokąty pokazują związek między zmiennymi, w przeciwieństwie do wykresu rozrzutu, nawet pomimo zastosowania bardzo małych punktów i ich losowego rozproszenia. Artykuł zawiera również kod, który został wykorzystany do stworzenia wykresów w środowisku R.

Słowa kluczowe: wykres rozrzutu, grafika, wizualizacja

Klasyfikacja AMS 2010: 62-09