Heat (and hexagon) plots in Stata

Ben Jann

University of Bern, ben.jann@soz.unibe.ch

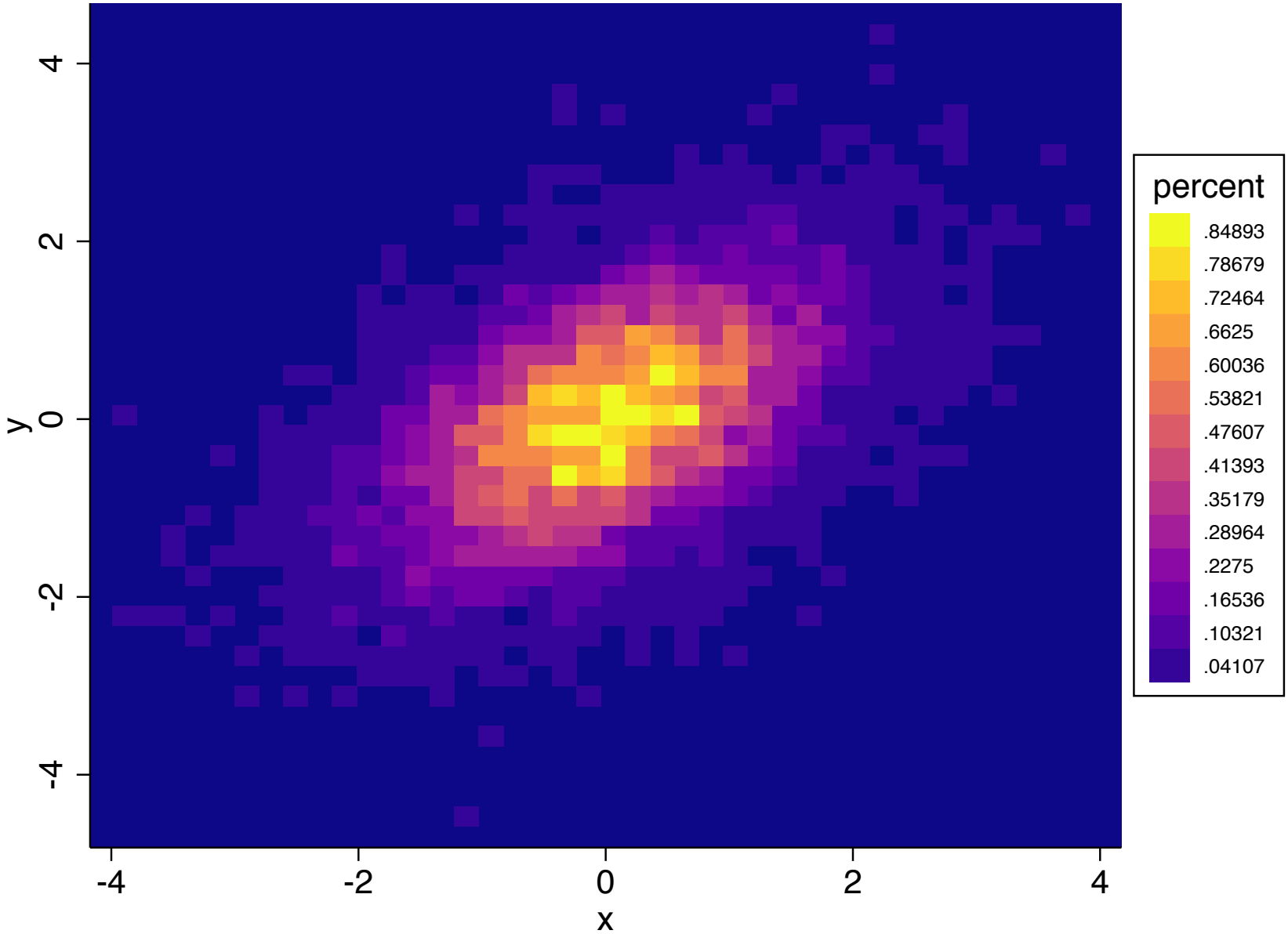
2019 London Stata Conference
London, September 5–6, 2019

Outline

What is a heat plot?

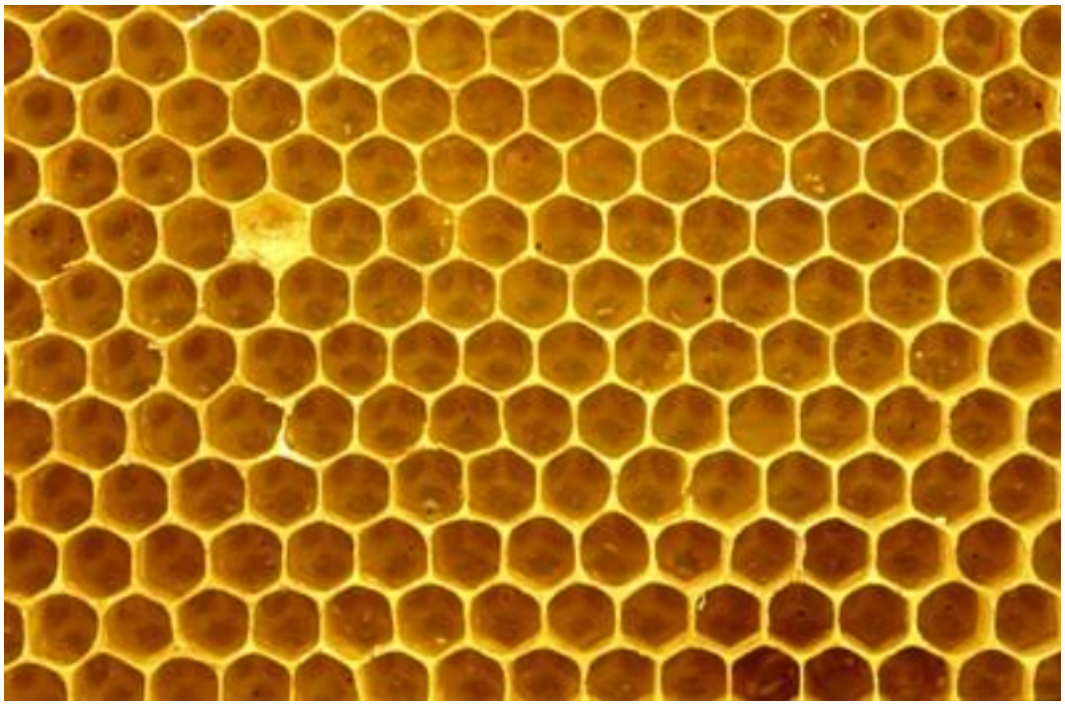
- Generally speaking, a heat plot is a graph in which some aspect of the data is displayed as a **color gradient**.
- A simple example is a **bivariate histogram**; the color gradient is used to illustrate (relative) frequencies within bins of X and Y.

```
. quietly drawnorm y x, n(10000) corr(1 .5 1) cstorage(lower) clear
. heatplot y x, backfill colors(plasma)
```



What about hexagons?

- Hexagons are great because they look a bit like circles, but you can join them together without leaving gaps.
- Bees found out how awesome hexagons are long time ago.



What about hexagons?

- Latter on, gully cover designers found out that hexagons look great on gully covers.

What about hexagons?

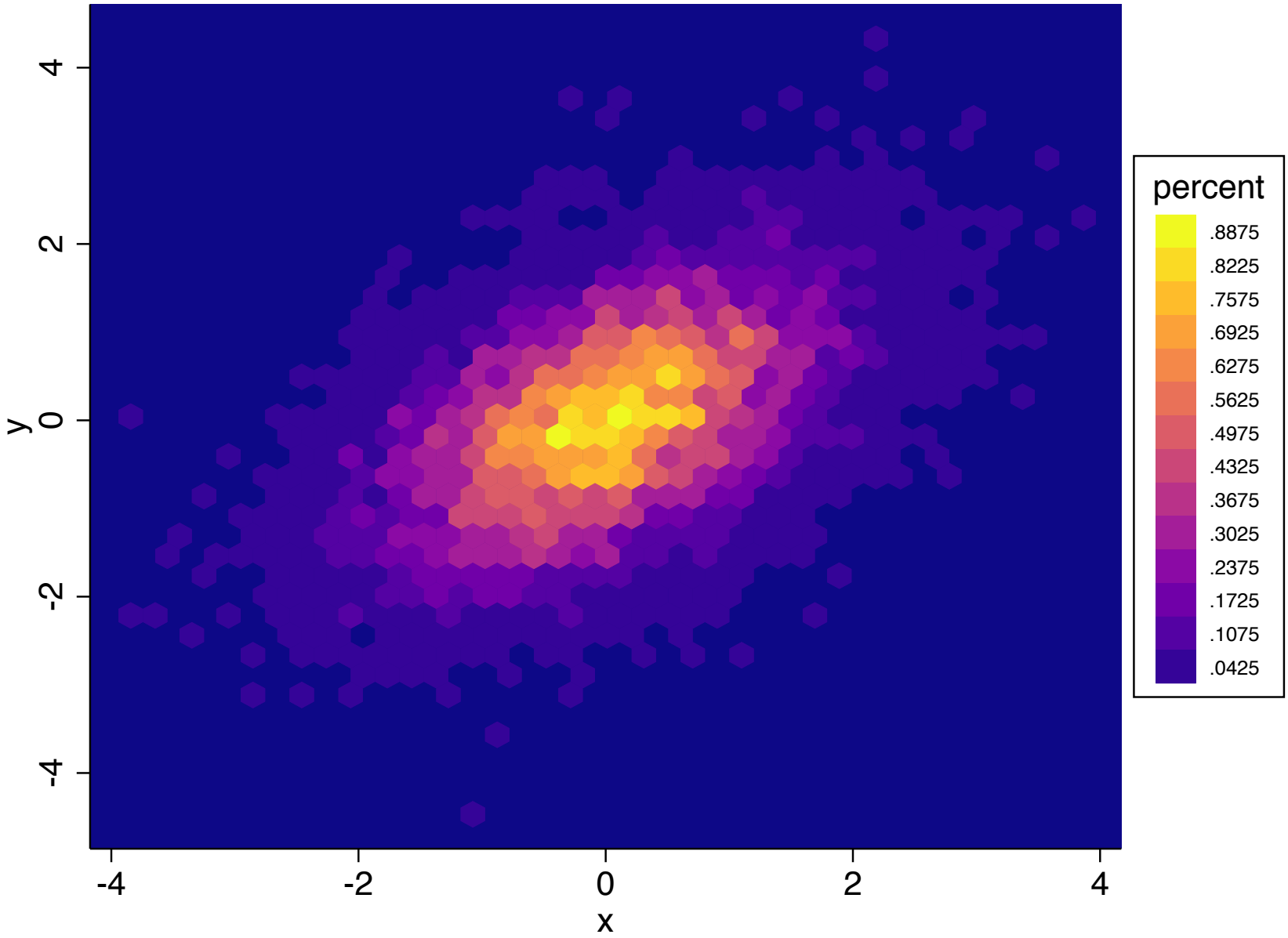
- Finally, also statisticians discovered the virtues of hexagons.

  "The here are many reasons for using hexagons, at least over squares. Hexagons have symmetry of nearest neighbors which is lacking in square bins. Hexagons are the maximum number of sides a polygon can have for a regular tesselation of the plane, so in terms of packing a hexagon is 13% more efficient for covering the plane than squares. This property translates into better sampling efficiency at least for elliptical shapes. Lastly hexagons are visually less biased for displaying densities than other regular tesselations. For instance with squares our eyes are drawn to the horizontal and vertical lines of the grid."[1]

[1] Lewin-Koh, N. (2018). Hexagon Binning: an Overview. Available from https://cran.r-project.org/web/packages/hexbin/vignettes/hexagon_binning.pdf

Example from above using hexagons

```
. hexplot y x, backfill colors(plasma)
```



Why heat plots (be it squares or hexagons)?

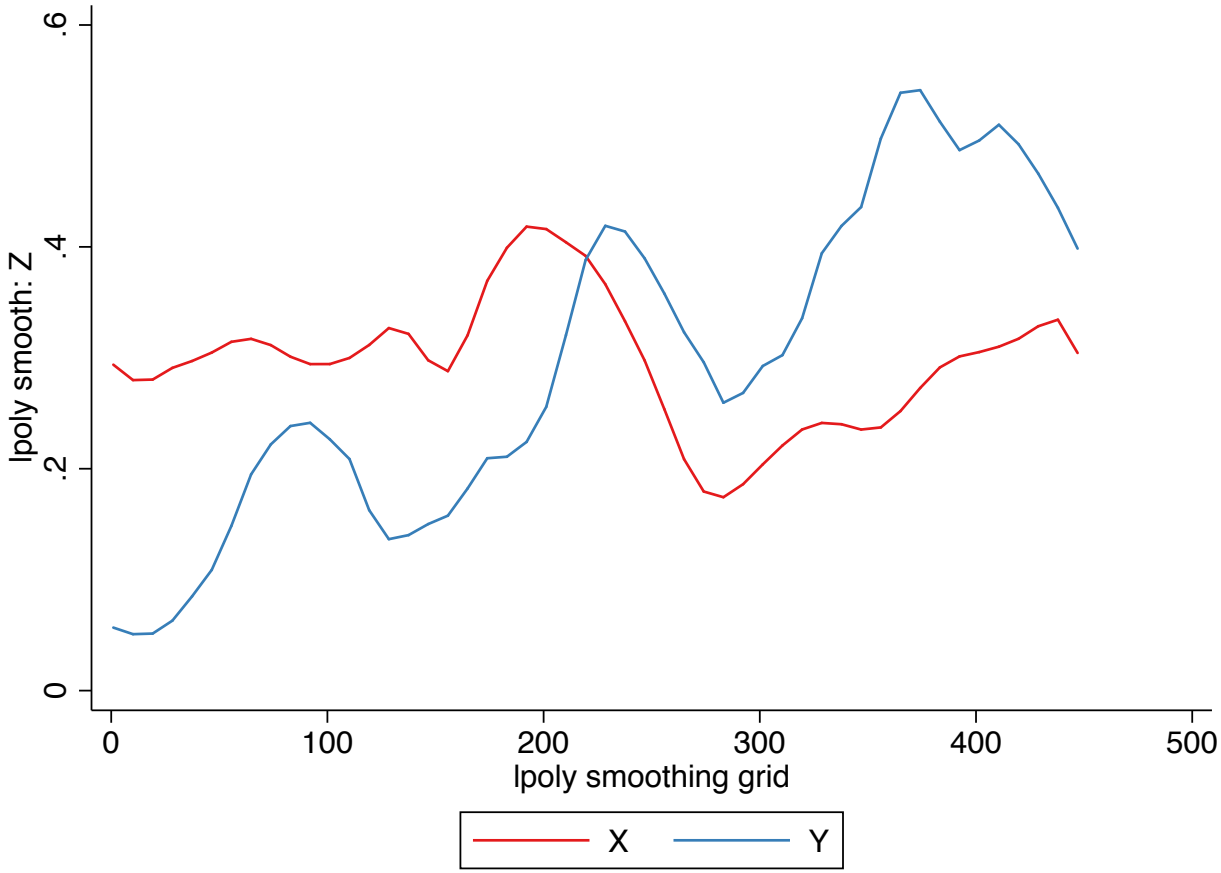
- Heat plots are great for visualizing structure in (large) datasets.
- Here is an example:

```
. use example, clear
. count
  134,100
. list in 1/10
```

	X	Y	Z
1.	16	193	.12484335
2.	371	13	.00772907
3.	157	380	.57315805
4.	334	443	.31666994
5.	424	205	.23699765
6.	47	319	.30675008
7.	50	288	.31003926
8.	434	5	.03925507
9.	180	303	.56515385
10.	428	183	.21671468

- Run some analyses . . .

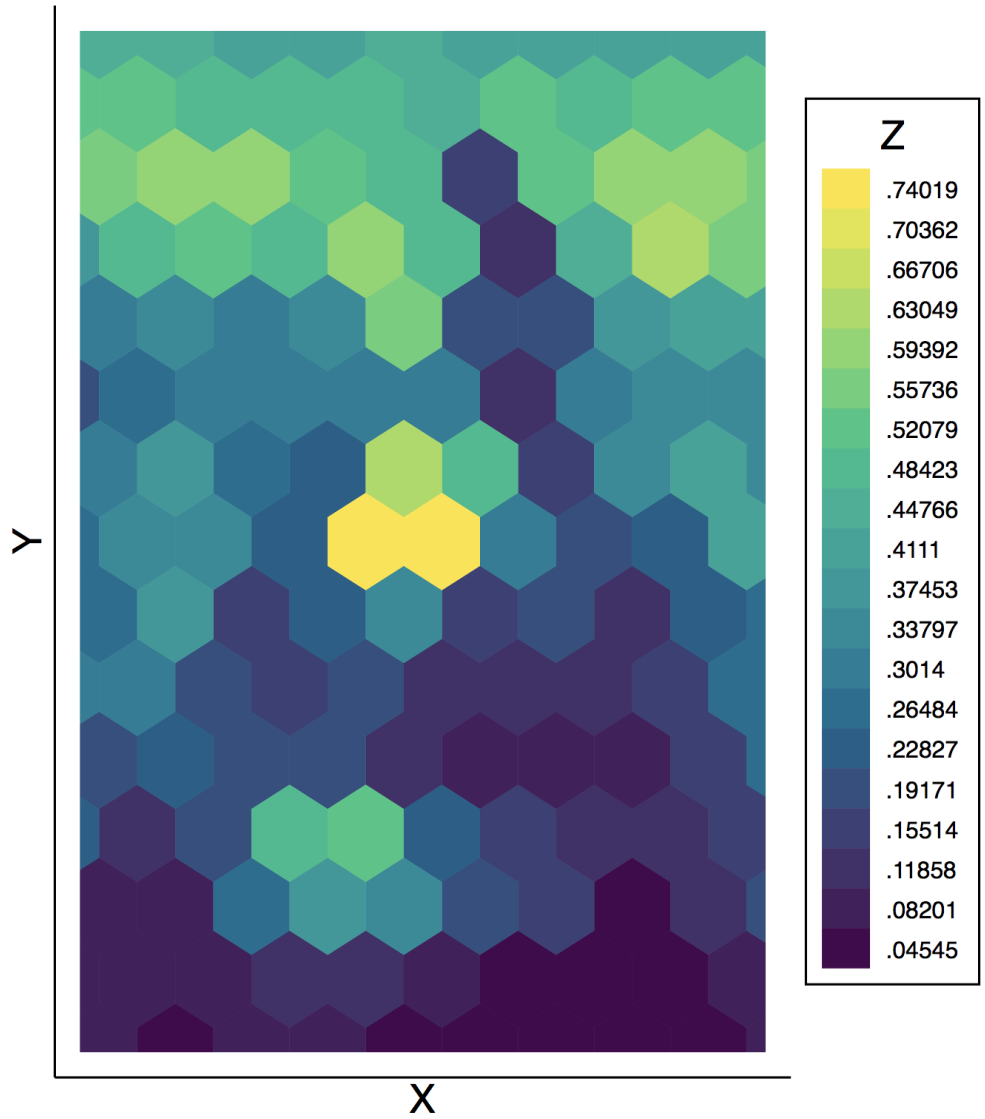
```
. two (lpoly Z X, degree(1)) (lpoly Z Y), legend(order(1 "X"  2 "Y"))
```



- Interesting! We clearly see the business cycles and a general upward trend in country Y, but country X did not develop much and there has been some severe crisis between time 200 and 300.

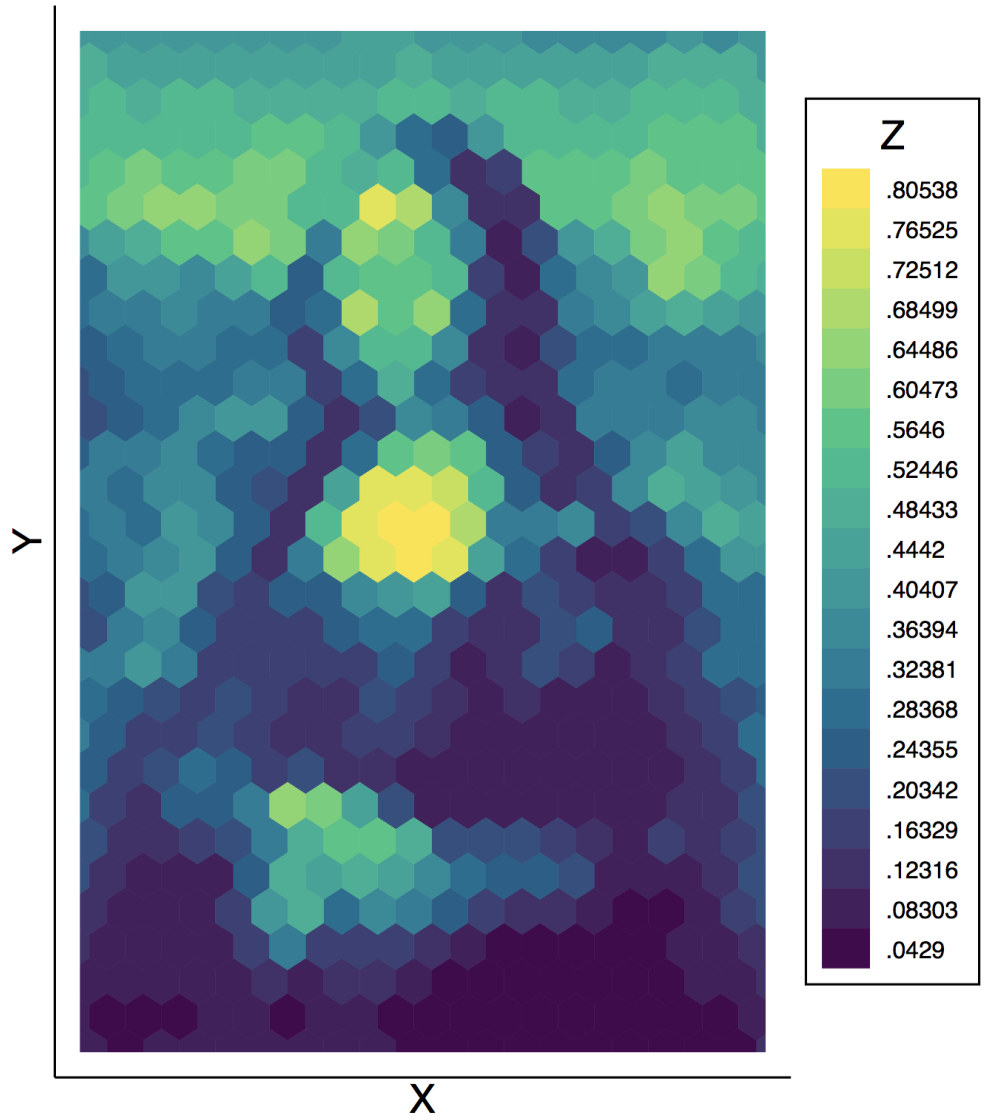
- Here is a heat plot of the data:

```
. hexplot Z Y X, xbins(10) ybins(15) levels(20) clip ///
>     xlabel(none) ylabel(none) aspect(`=447/300')
```



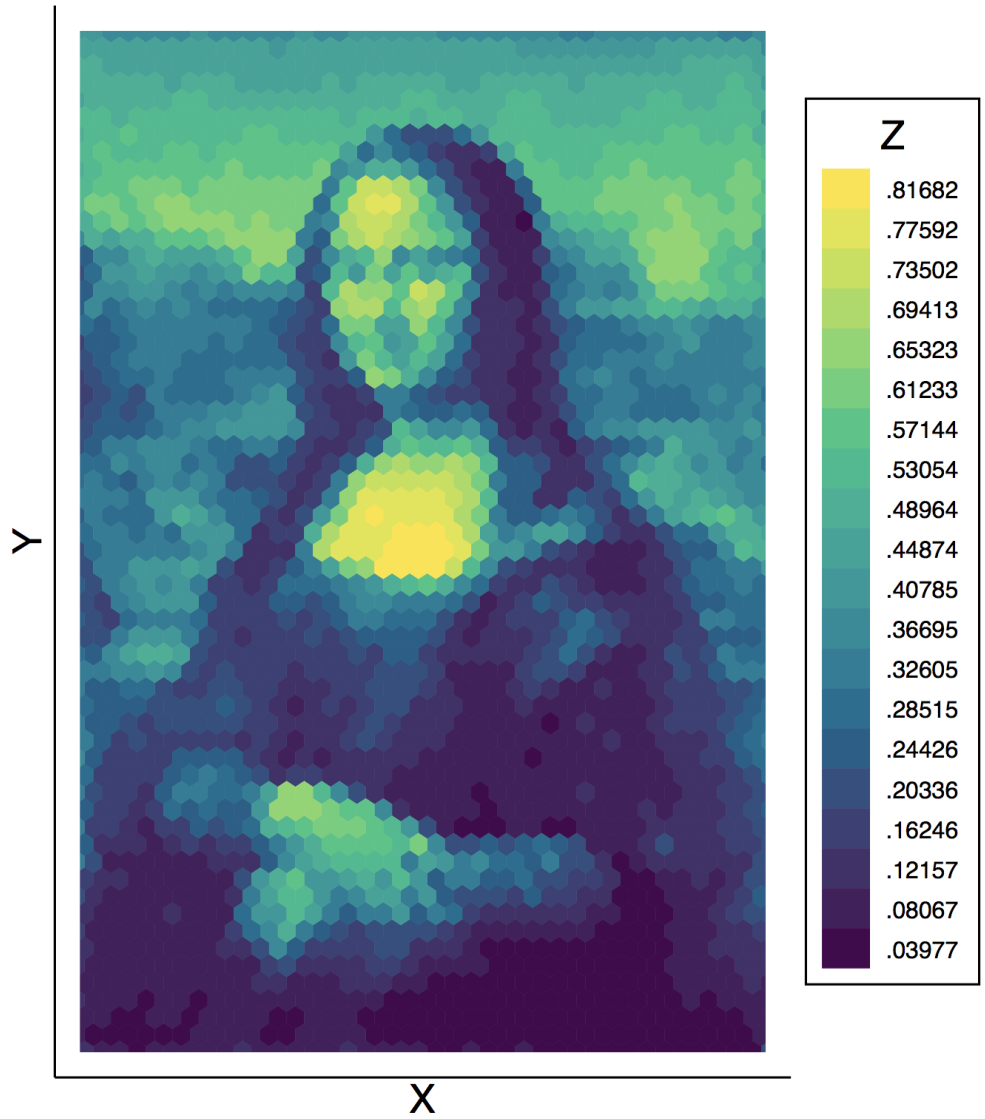
- Here is a heat plot of the data:

```
. hexplot Z Y X, xbins(20) ybins(30) levels(20) clip ///
>     xlabel(none) ylabel(none) aspect(`=447/300')
```



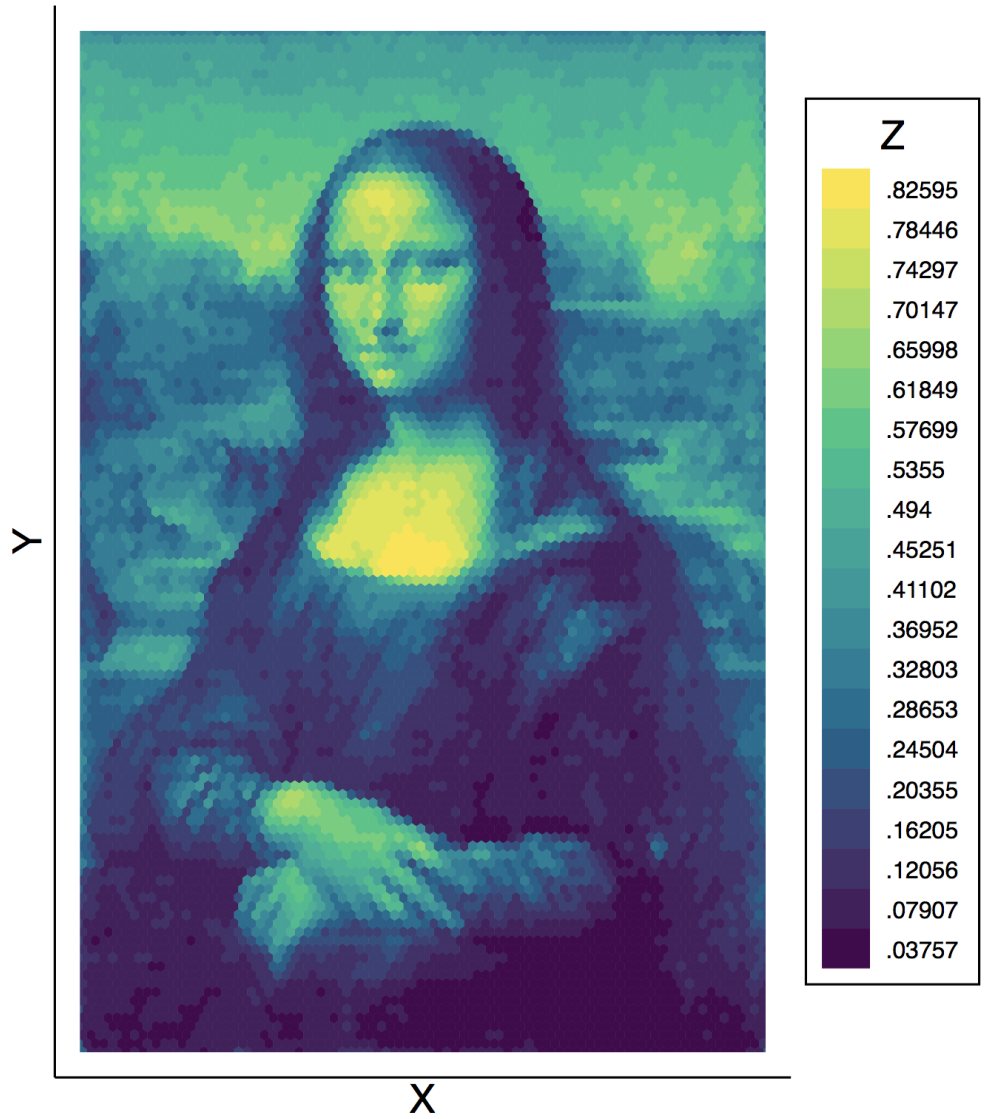
- Here is a heat plot of the data:

```
. hexplot Z Y X, xbins(40) ybins(60) levels(20) clip ///
>     xlabel(none) ylabel(none) aspect(`=447/300')
```



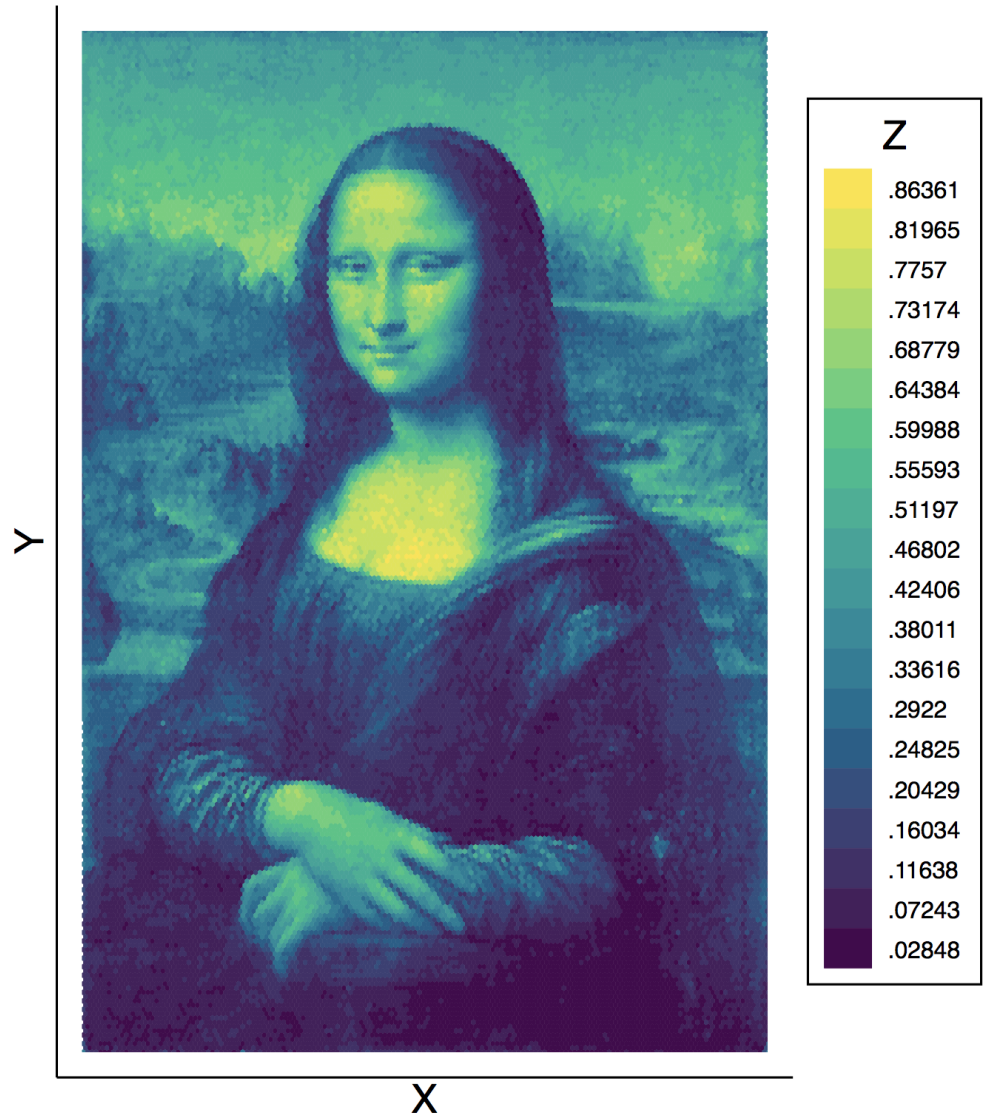
- Here is a heat plot of the data:

```
. hexplot Z Y X, xbins(80) ybins(120) levels(20) clip ///
>     xlabel(none) ylabel(none) aspect(`=447/300')
```



- Here is a heat plot of the data:

```
. hexplot Z Y X, xbins(160) ybins(240) levels(20) clip ///
>     xlabel(none) ylabel(none) aspect(`=447/300')
```



Main commands

- Bivariate histogram

  `heatplot` *Y X* [*if*] [*in*] [*weight*] [, *options*]

- Trivariate heat plot (color gradient for *Z*)

  `heatplot` *Z Y X* [*if*] [*in*] [*weight*] [, *options*]

- Heat plot from Stata matrix

  `heatplot` *matname* [, *options*]

- Heat plot from Mata matrix

  `heatplot mata(`*name*`)` [, *options*]

- Heat plot using hexagons

  `hexplot ...`

Main options

- Color gradient options

`levels(`#`)`	number of color bins
`cuts(`*numlist*`)`	custom cutpoints for color bins
`colors(`*palette*`)`	color map to be used for the color bins
`statistic(`*stat*`)`	how Z is aggregated
`size`[`(`*exp*`)`] | `sizeprop`	size of color fields
`values(`*options*`)`	display values as marker labels
`scatter`[`(...)`]	render color fields as scatter plot
`keylabels(`*spec*`)`	how legend keys are labeled

...

- Binning of Y and X

[`x`|`y`]`bins(`*spec*`)`	how continuous Y and X are binned
[`x`|`y`]`bwidth(`*spec*`)`	alternative to `bins()`
[`x`|`y`]`discrete`[`(`#`)`]	treat variables as discrete and omit binning

(note: categorical X and Y can be specified as `i.`*varname*)

...

Main options

- Matrix options

`drop(`*numlist*`)`	drop elements equal to values in *numlist*
`lower`	display lower triangle only
`upper`	display upper triangle only
...	

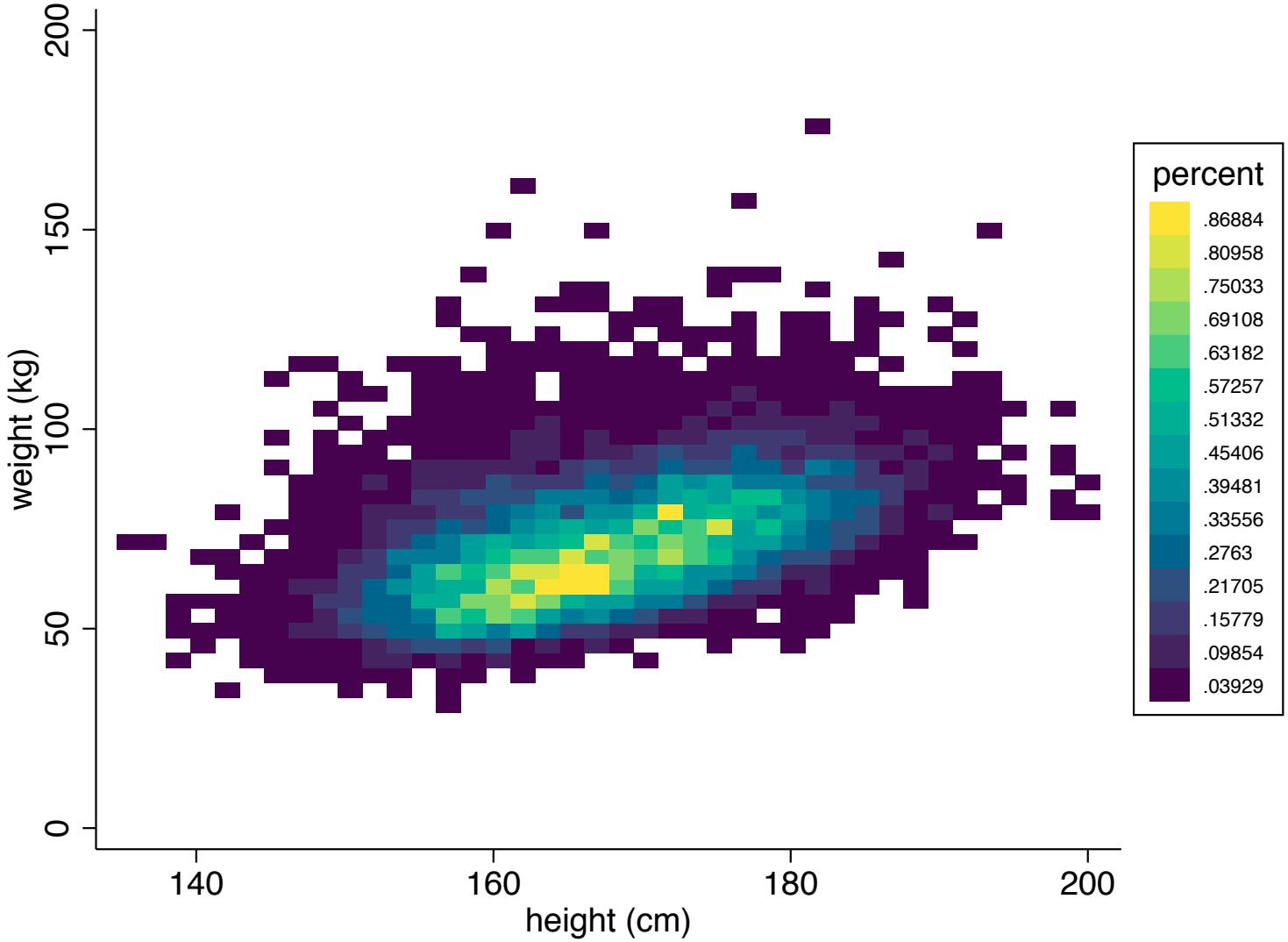
- Graph options

`addplot(`*plots*`)`	add other plots to the graph
`by(`*varlist* [`,` *options*]`)`	repeat plot by subgroups
twoway_options	general twoway options
...	

- Some more options related to storing results ...

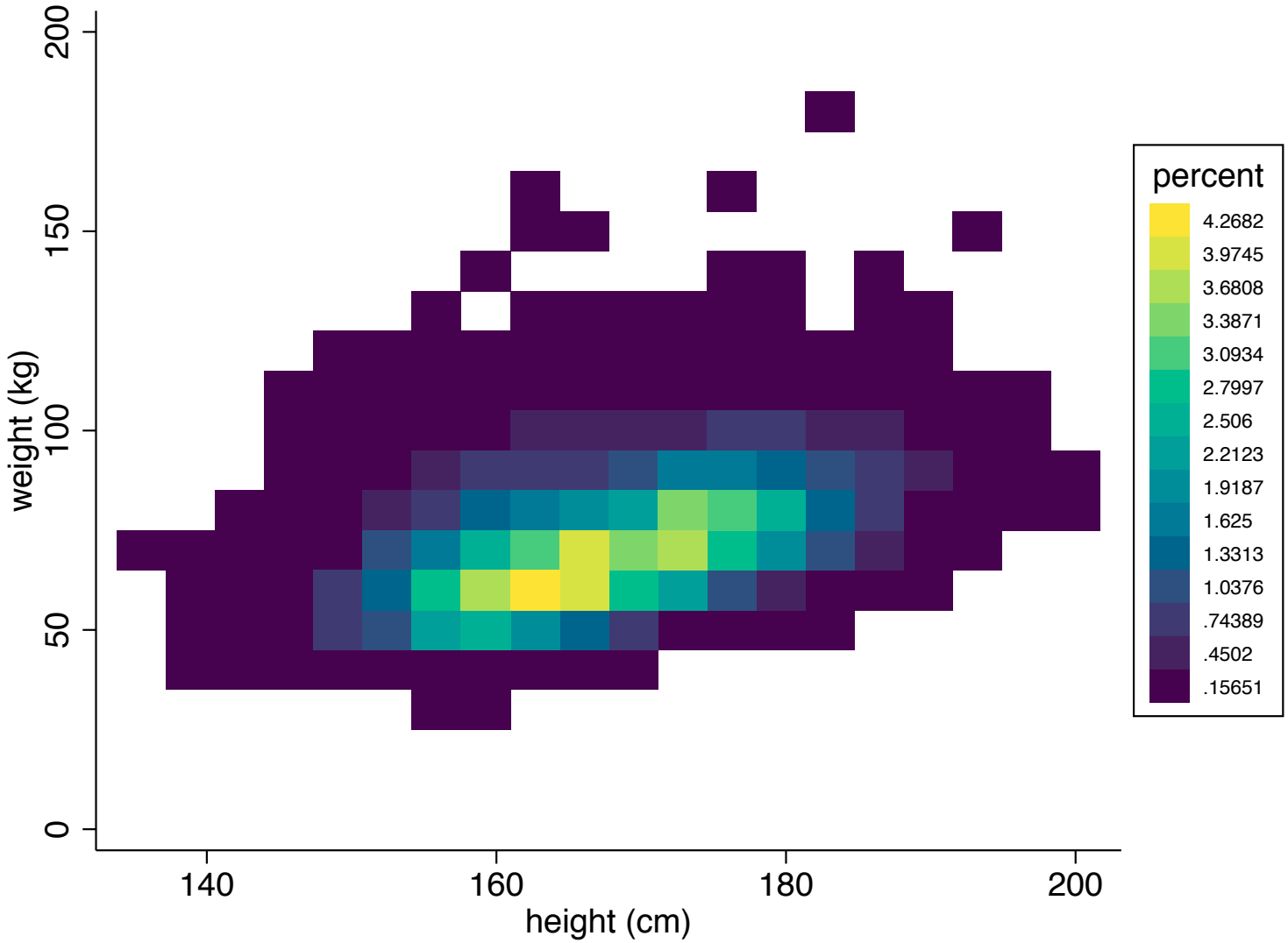
Default

```
. webuse nhanes2, clear
. heatplot weight height
```



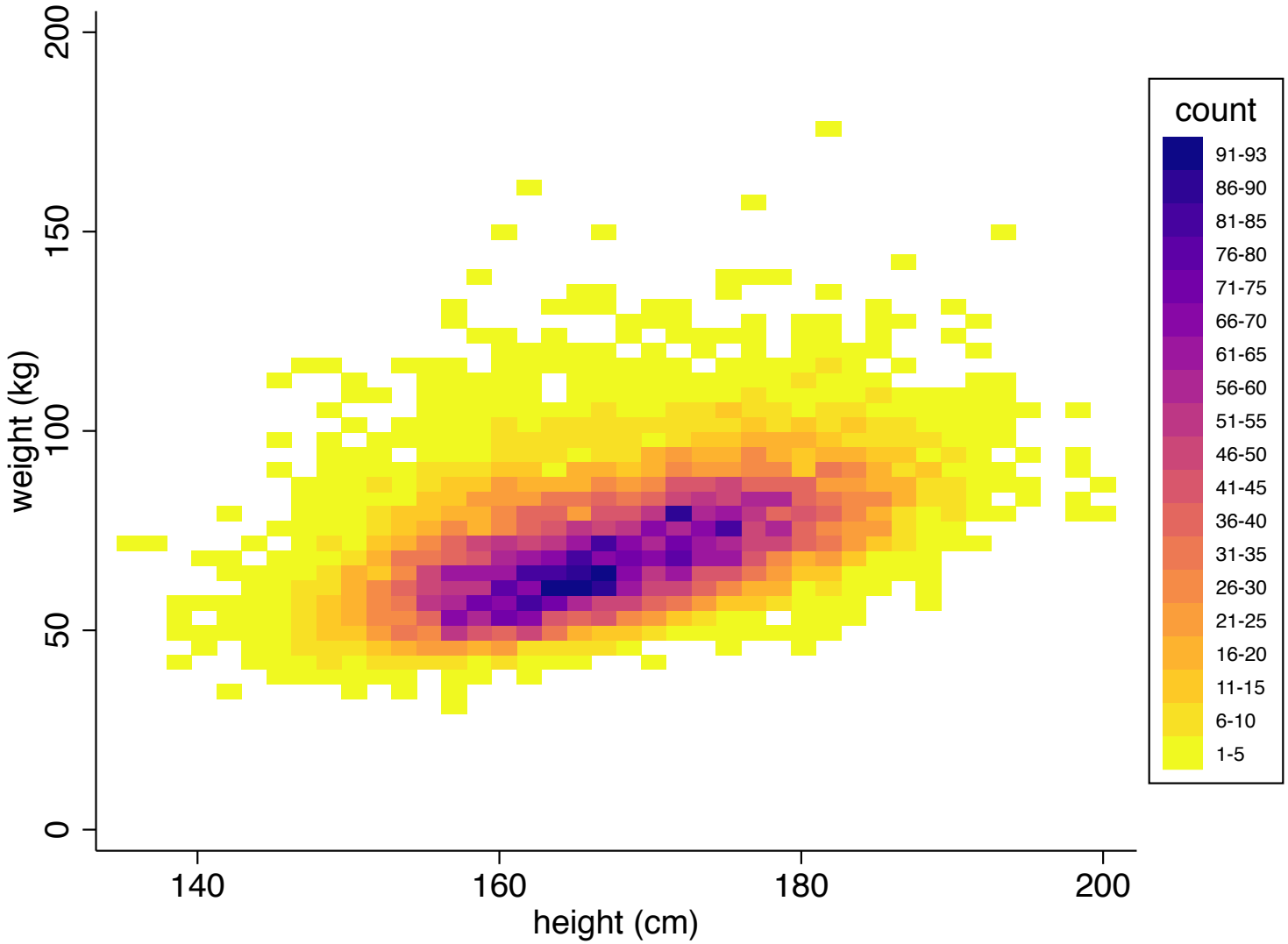
Change resolution

```
. heatplot weight height, xbins(20) ybwidth(10 30)
```



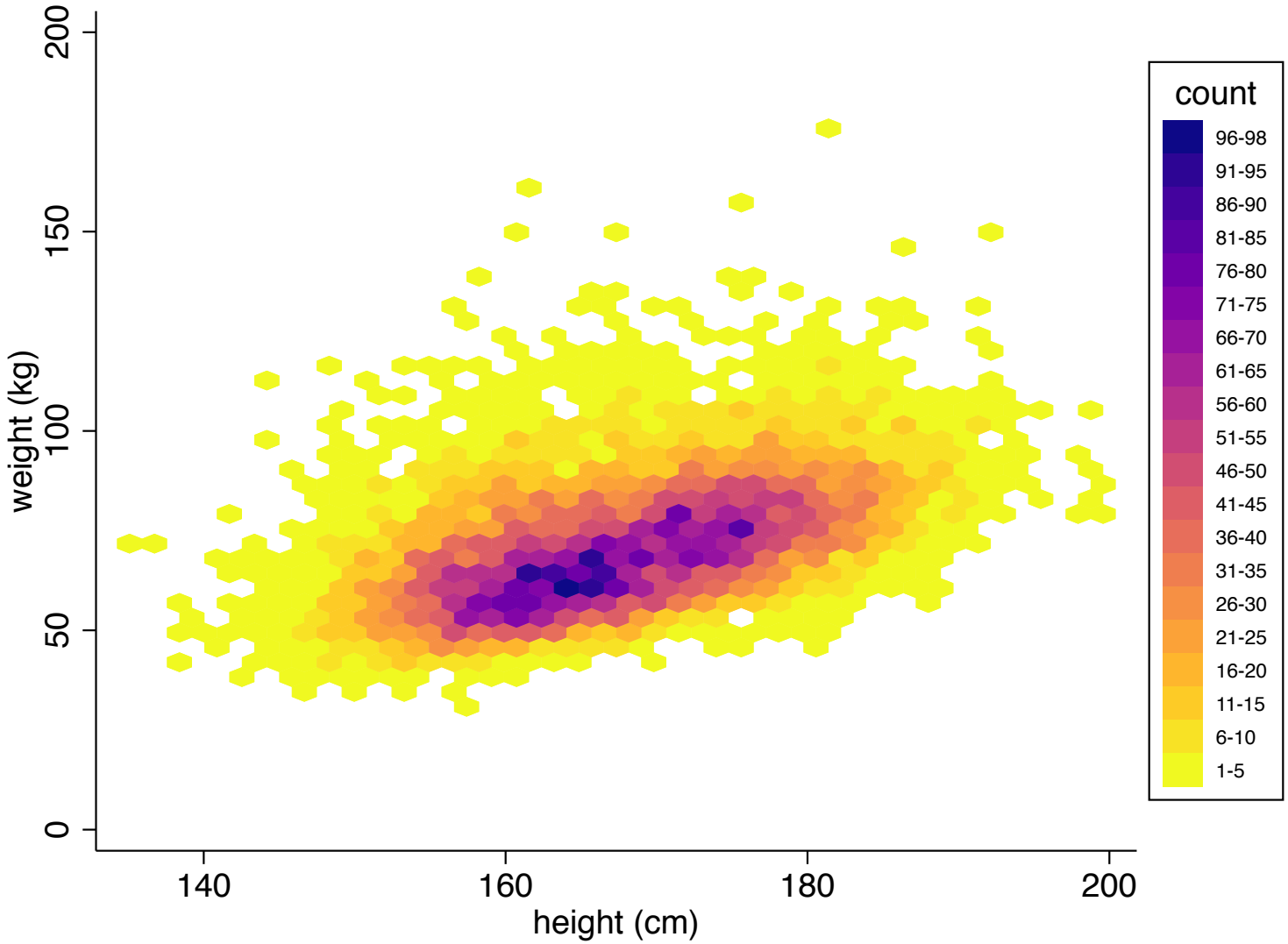
Use counts, change color ramp, change binning, and labeling

```
. heatplot weight height, statistic(count) color(plasma, reverse) ///
>     cut(1(5)@max) keylabels(, range(1))
```



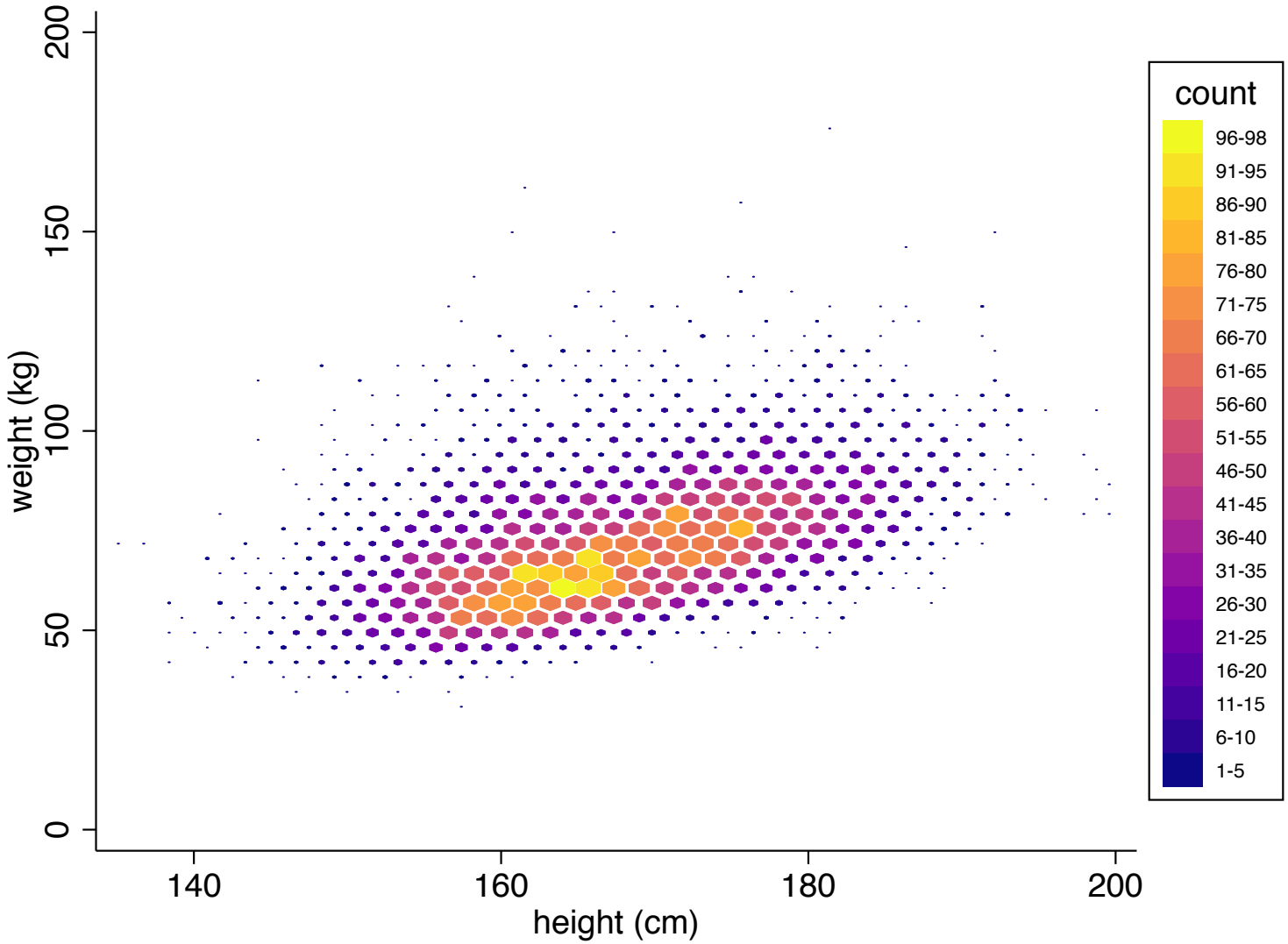
Use hexagons instead of squares

```
. hexplot weight height, statistic(count) color(plasma, reverse) ///
>     cut(1(5)@max) keylabels(, range(1))
```



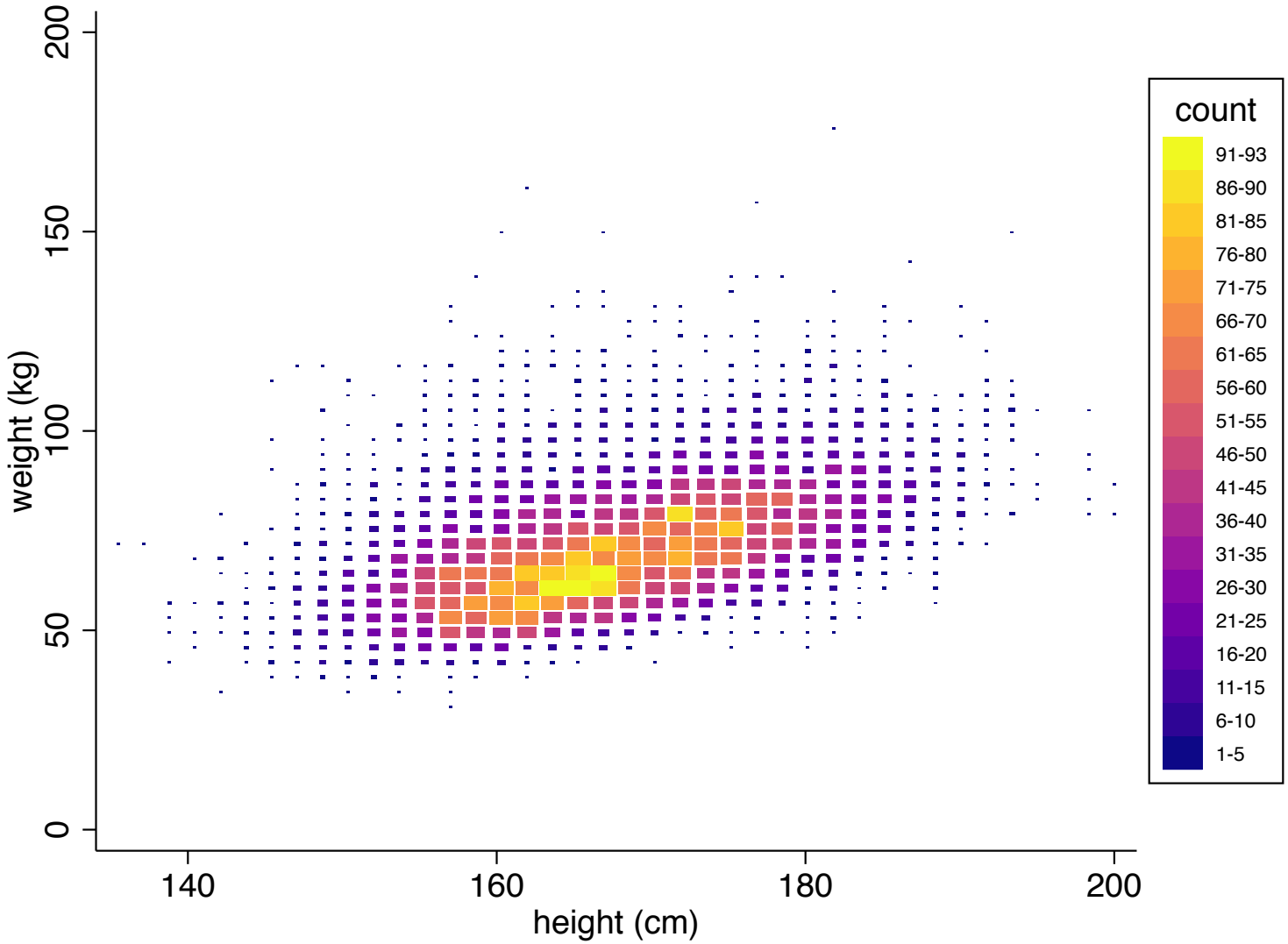
Scale size of hexagons by relative frequency

```
. hexplot weight height, statistic(count) color(plasma) ///
>     cut(1(5)@max) keylabels(, range(1)) size
```



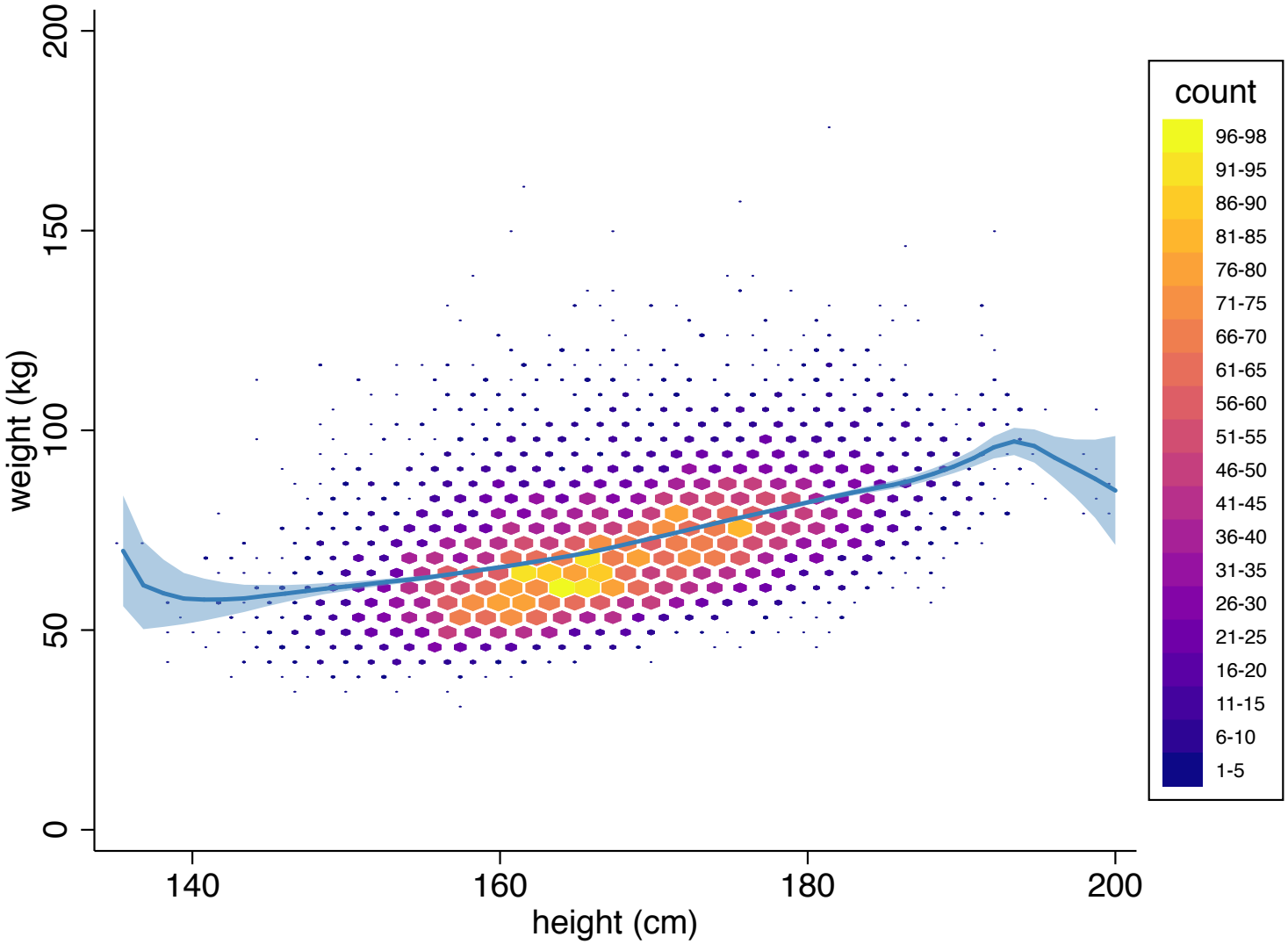
Scaling also available with squares

```
. heatplot weight height, statistic(count) color(plasma) ///
>     cut(1(5)@max) keylabels(, range(1)) size
```



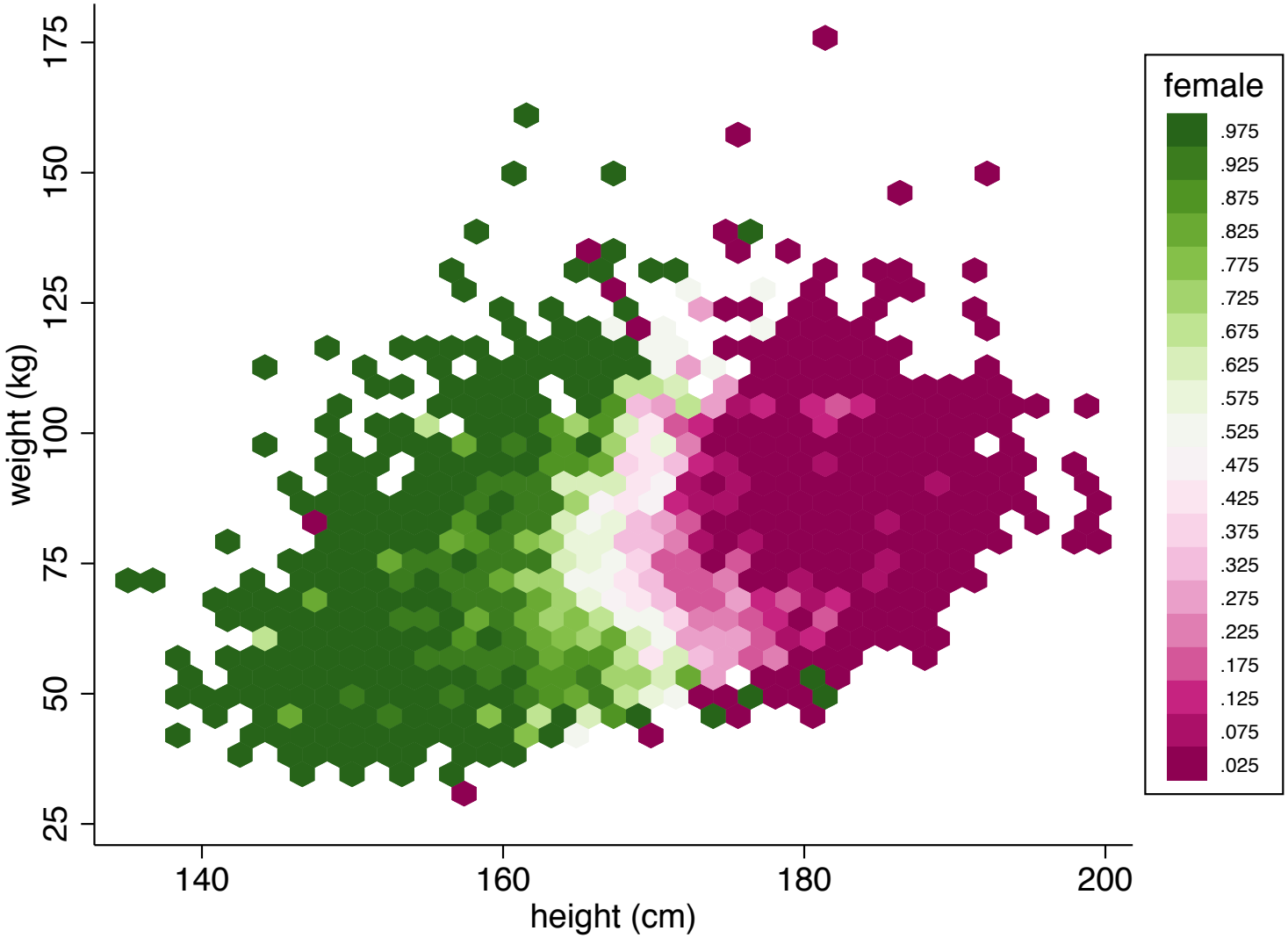
Adding other plots

```
. hexplot weight height, statistic(count) color(plasma) ///
>     cut(1(5)@max) keylabels(, range(1)) size ///
>     addplot(lpolyci weight height, degree(1) psty(p2) lw(*1.5) ac(%50) alc(%0))
```



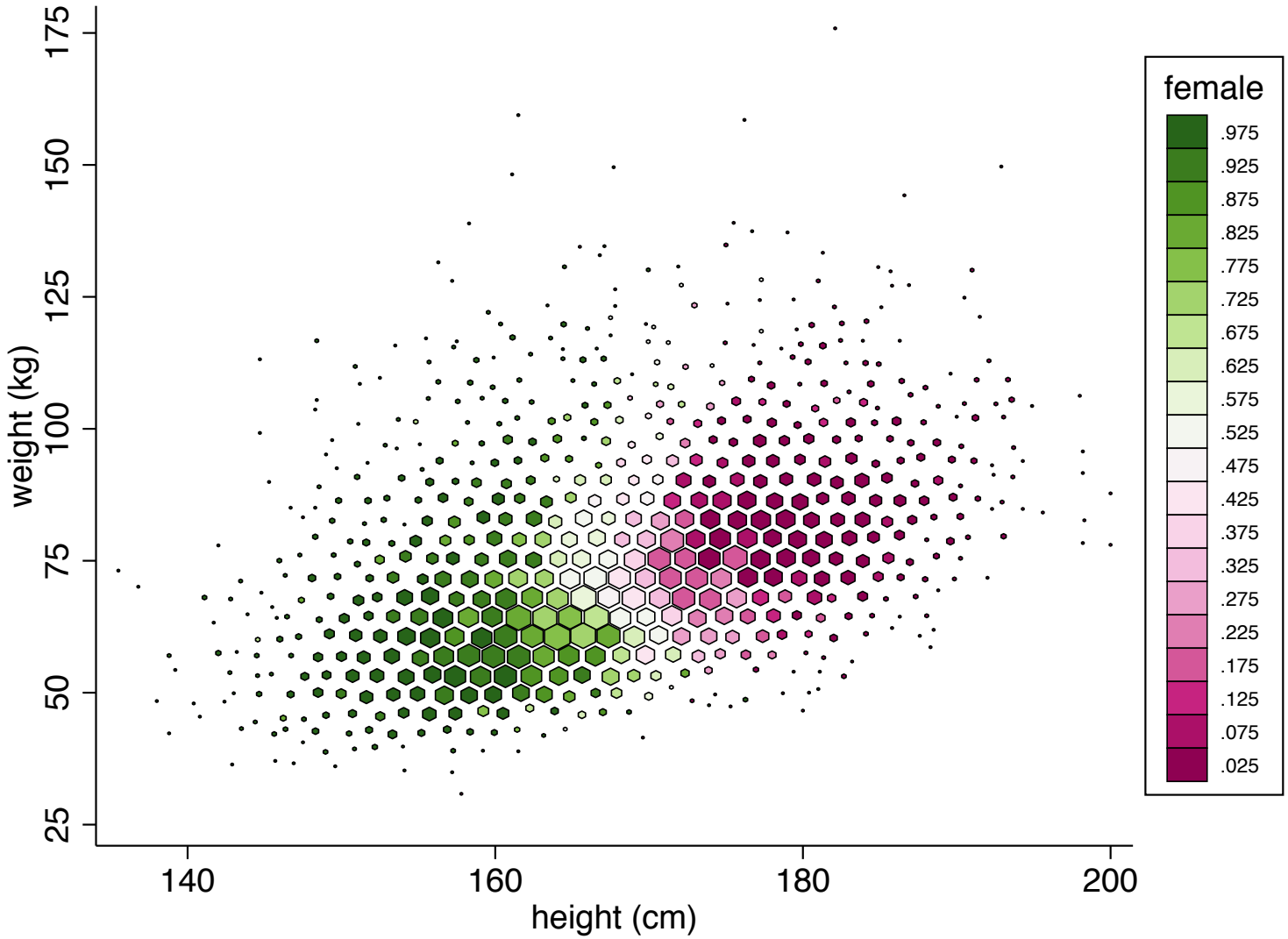
Gender distribution (proportion female) by weight and height

```
. webuse nhanes2, clear
. hexplot female weight height, color(PiYG) ylabel(25(25)175) cuts(0(.05)1)
```



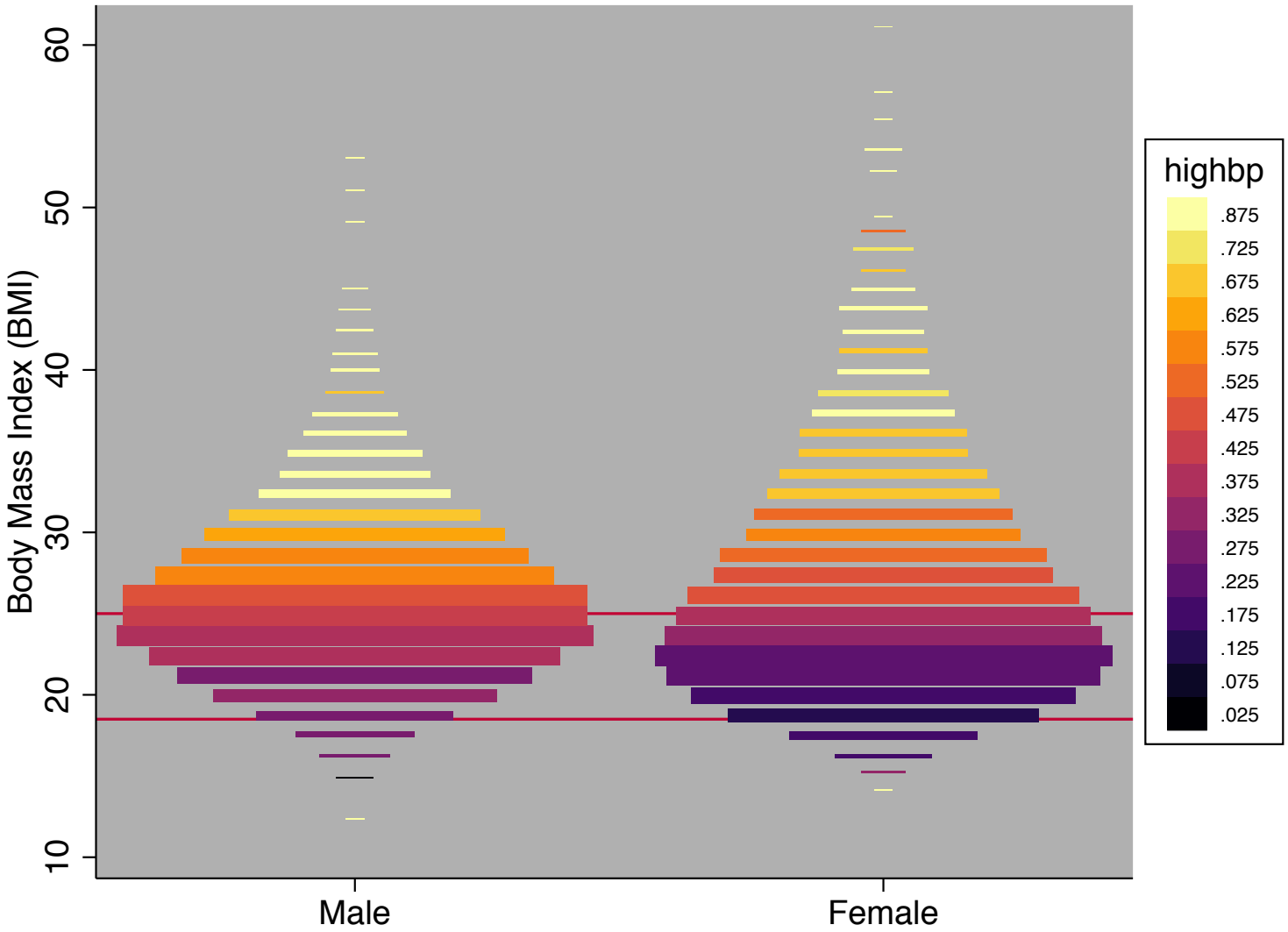
Same graph taking account relative frequencies

```
. hexplot female weight height, color(PiYG) ylabel(25(25)175) cuts(0(.05)1) ///
>     sizeprop recenter p(lcolor(black) lwidth(vthin) lalign(center))
```



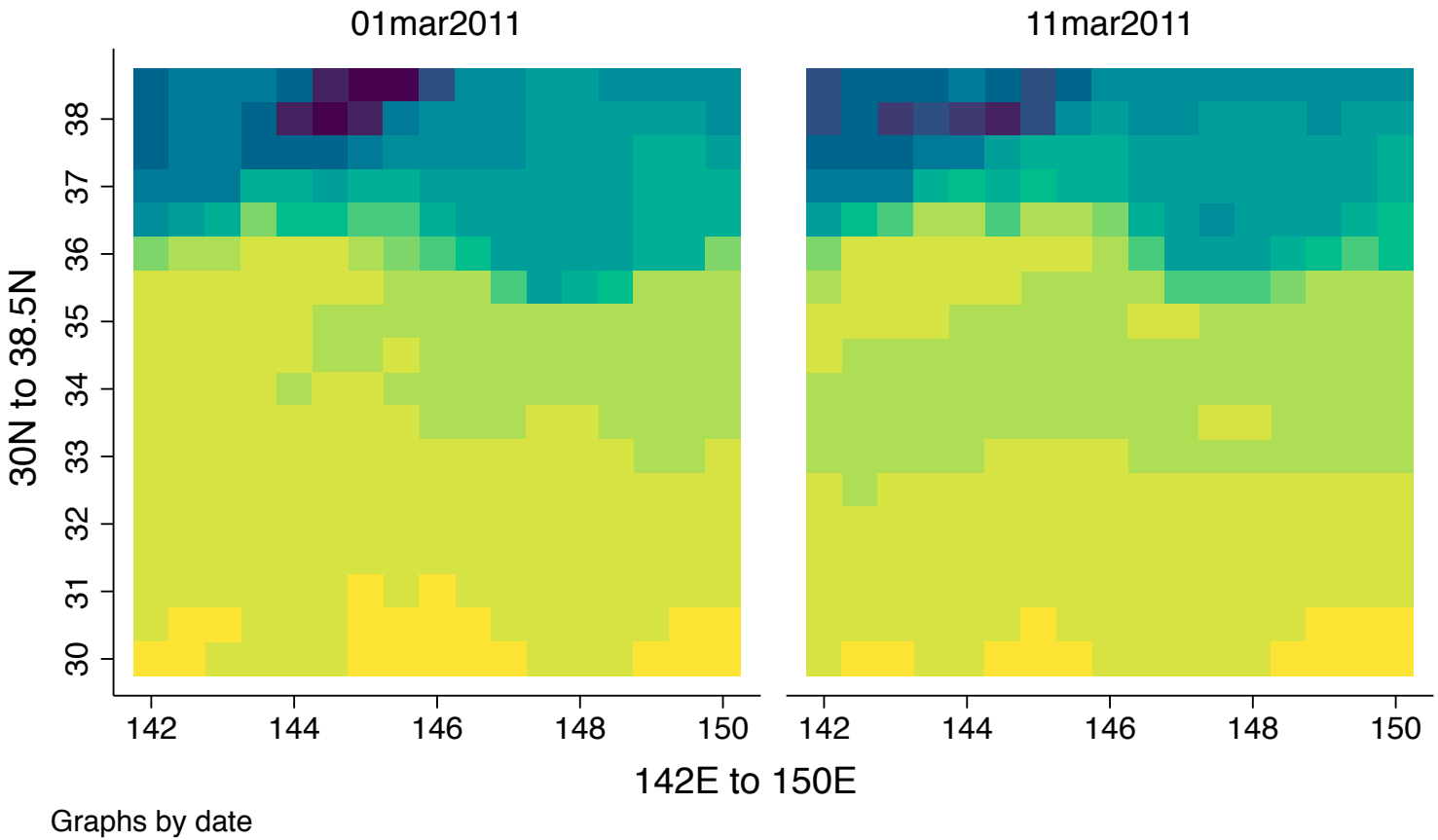
Distribution of the body mass index by gender and its relation to high blood pressure

```
. heatplot highbp bmi i.sex, xdiscrete(0.9) yline(18.5 25) cuts(0(.05).75) ///
>     sizeprop recenter colors(inferno) plotregion(color(gs11)) ylabel(, nogrid)
```



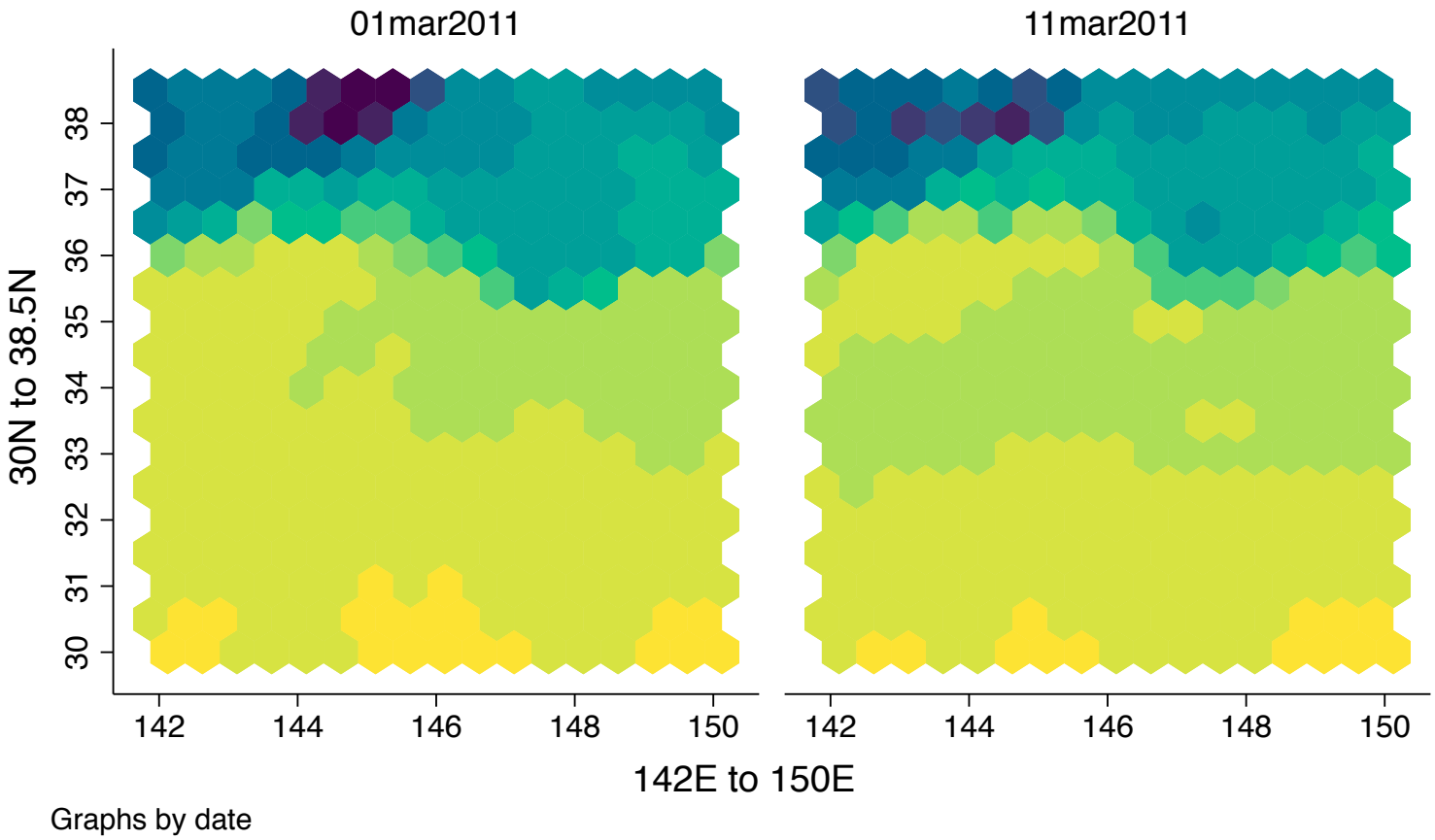
Sea surface temperature by longitude, latitude, and date

```
. sysuse surface, clear
(NOAA Sea Surface Temperature)
. heatplot temperature longitude latitude, discrete(.5) statistic(asis) ///
>     by(date, legend(off)) ylabel(30(1)38) aspectratio(1)
```



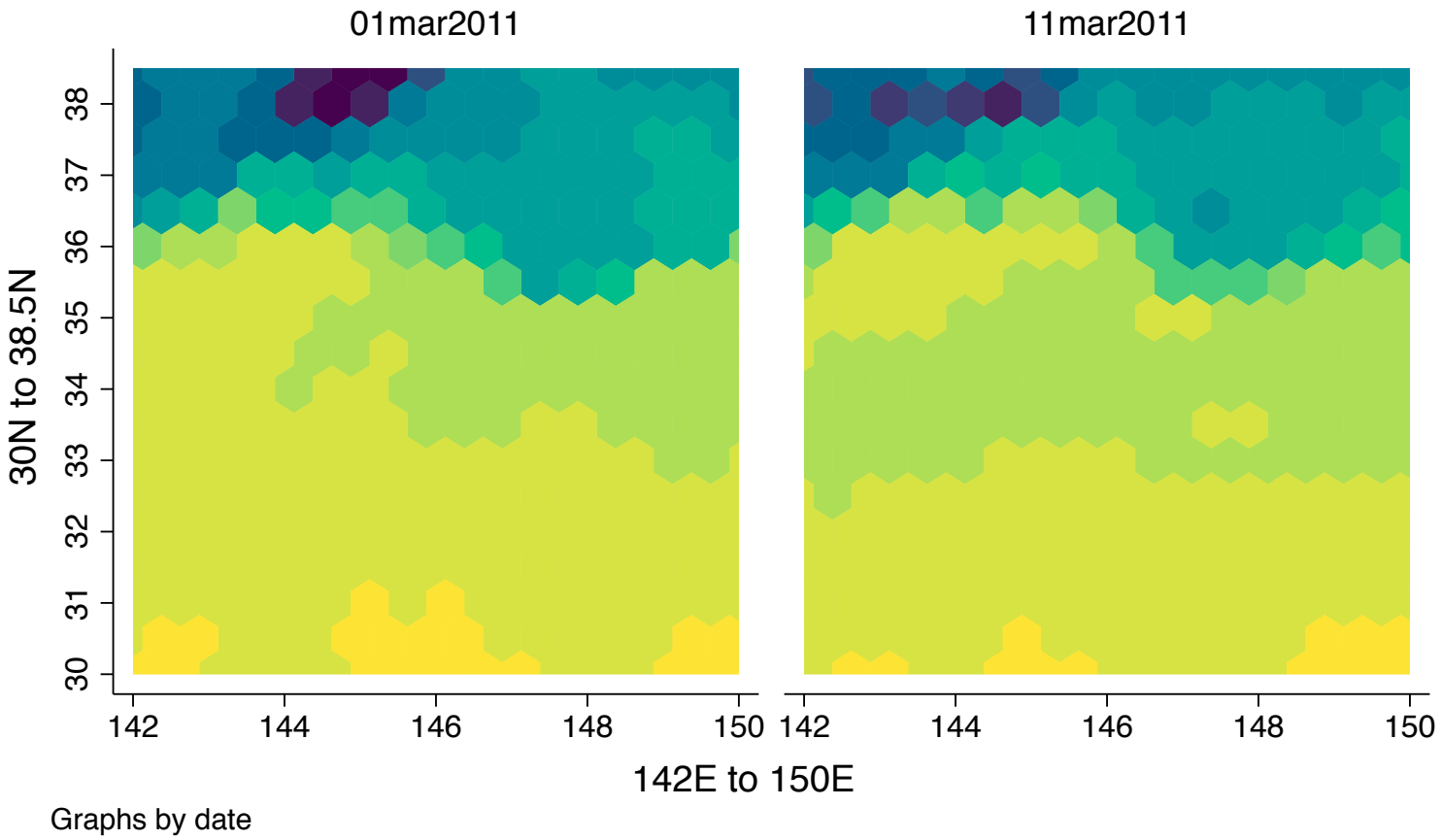
Same plot using hexagons

```
. hexplot temperature longitude latitude, discrete(.5) statistic(asis) ///
>     by(date, legend(off)) ylabel(30(1)38) aspectratio(1)
```



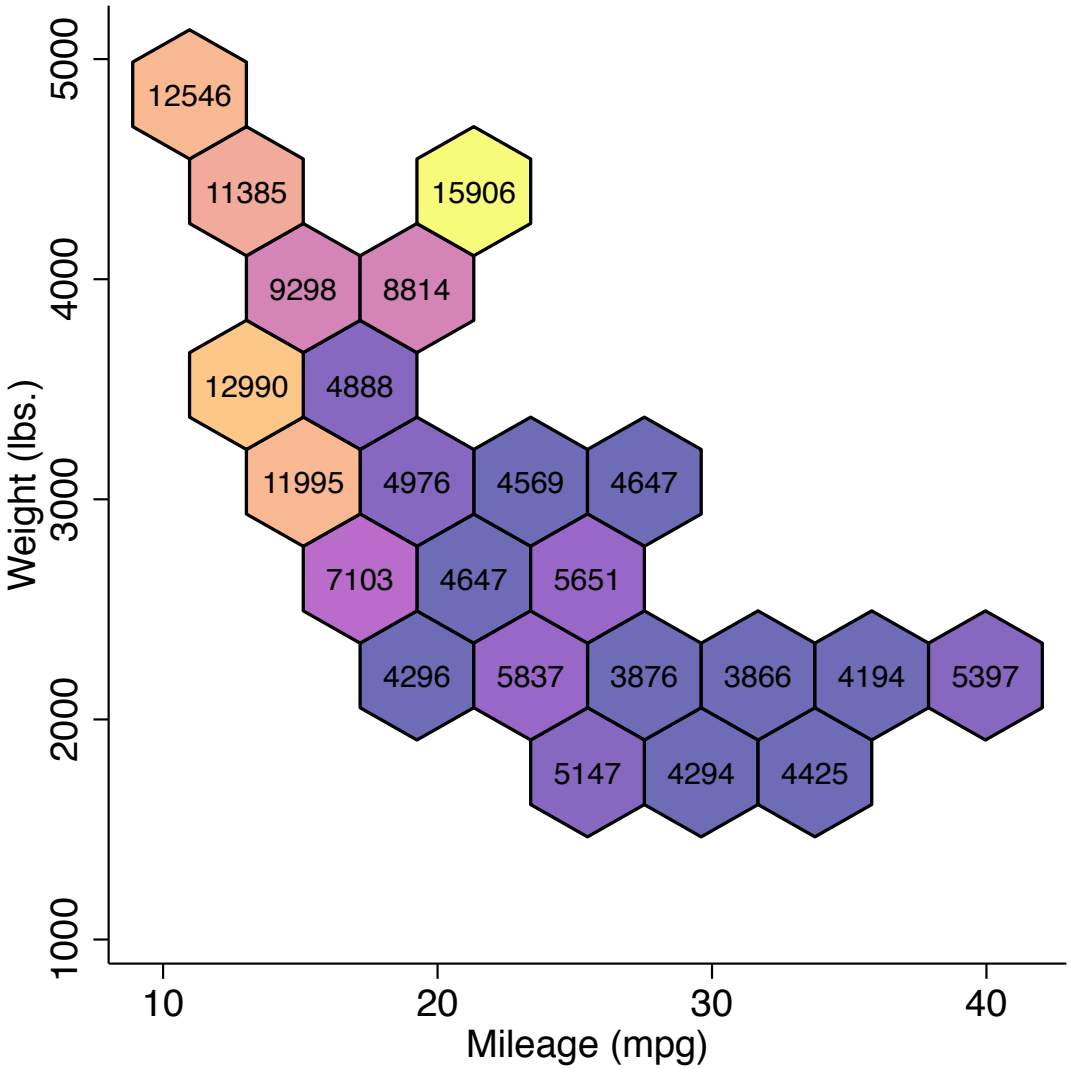
Same plot using hexagons

```
. hexplot temperature longitude latitude, discrete(.5) statistic(asis) clip ///
>     by(date, legend(off)) ylabel(30(1)38) aspectratio(1)
```



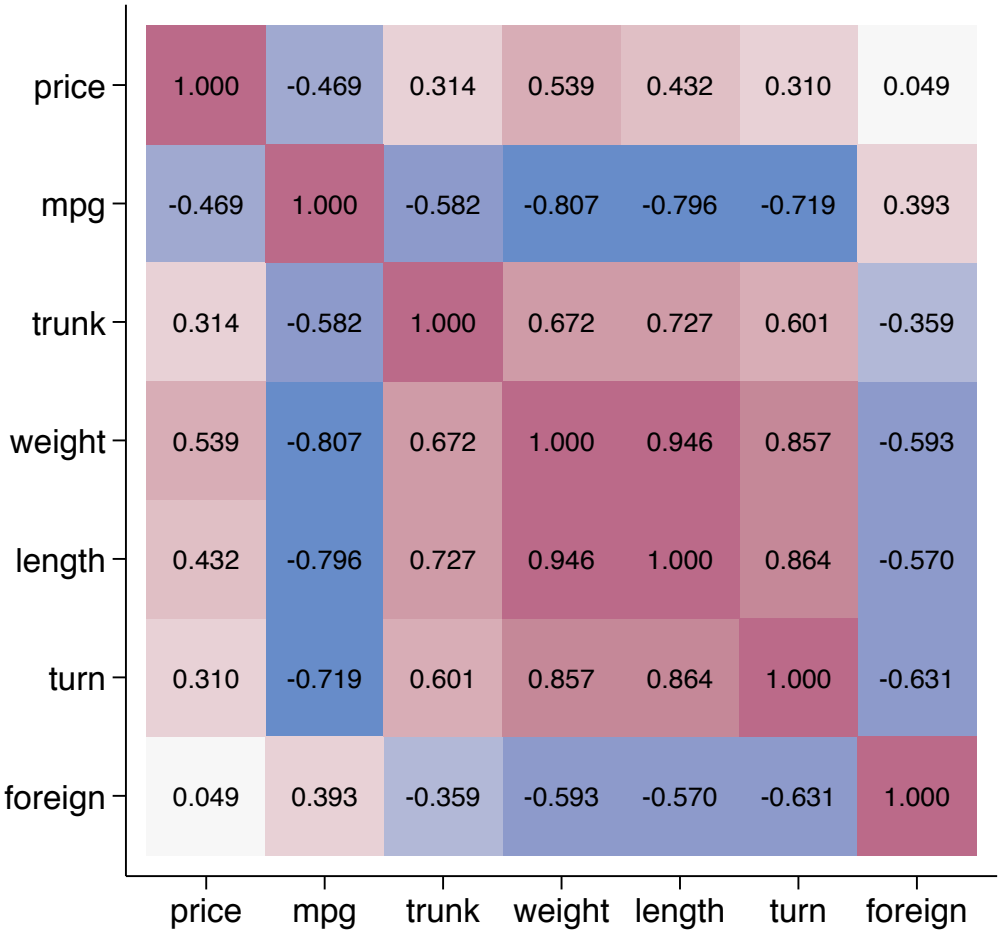
Same plot using hexagons

```
. quietly sysuse auto, clear
. hexplot price weight mpg, values(format(%9.0f)) legend(off) aspectratio(1) ///
>     colors(plasma, intensity(.6)) p(lc(black) lalign(center))
```



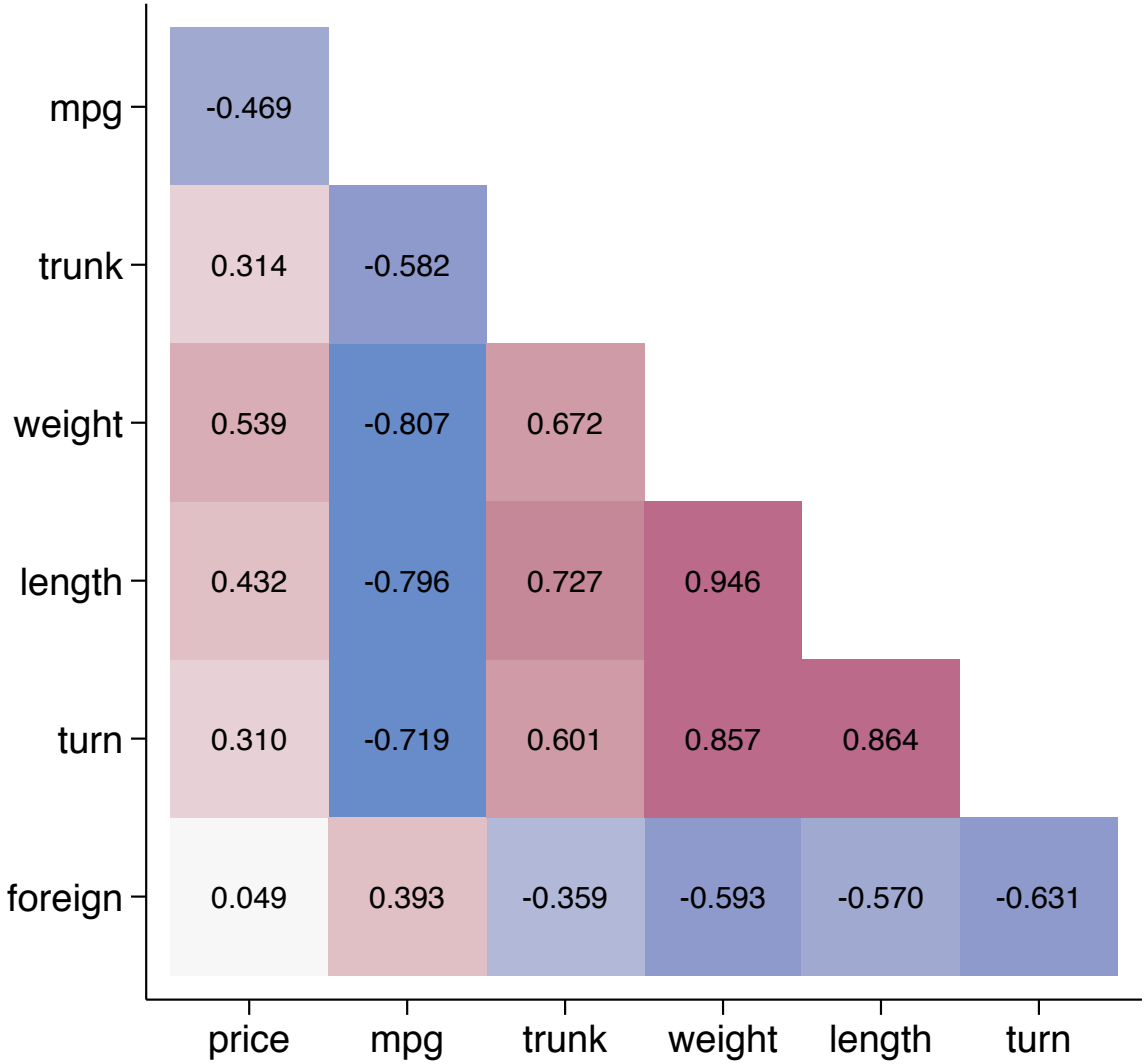
First store correlations in a matrix and then plot from there

```
. quietly sysuse auto, clear
. quietly correlate price mpg trunk weight length turn foreign
. matrix C = r(C)
. heatplot C, values(format(%9.3f)) color(hcl, diverging intensity(.6)) ///
>     legend(off) aspectratio(1)
```



Plot lower triangle only

```
. heatplot C, values(format(%9.3f)) color(hcl, diverging intensity(.6)) ///
>     legend(off) aspectratio(1) lower nodiagonal
```

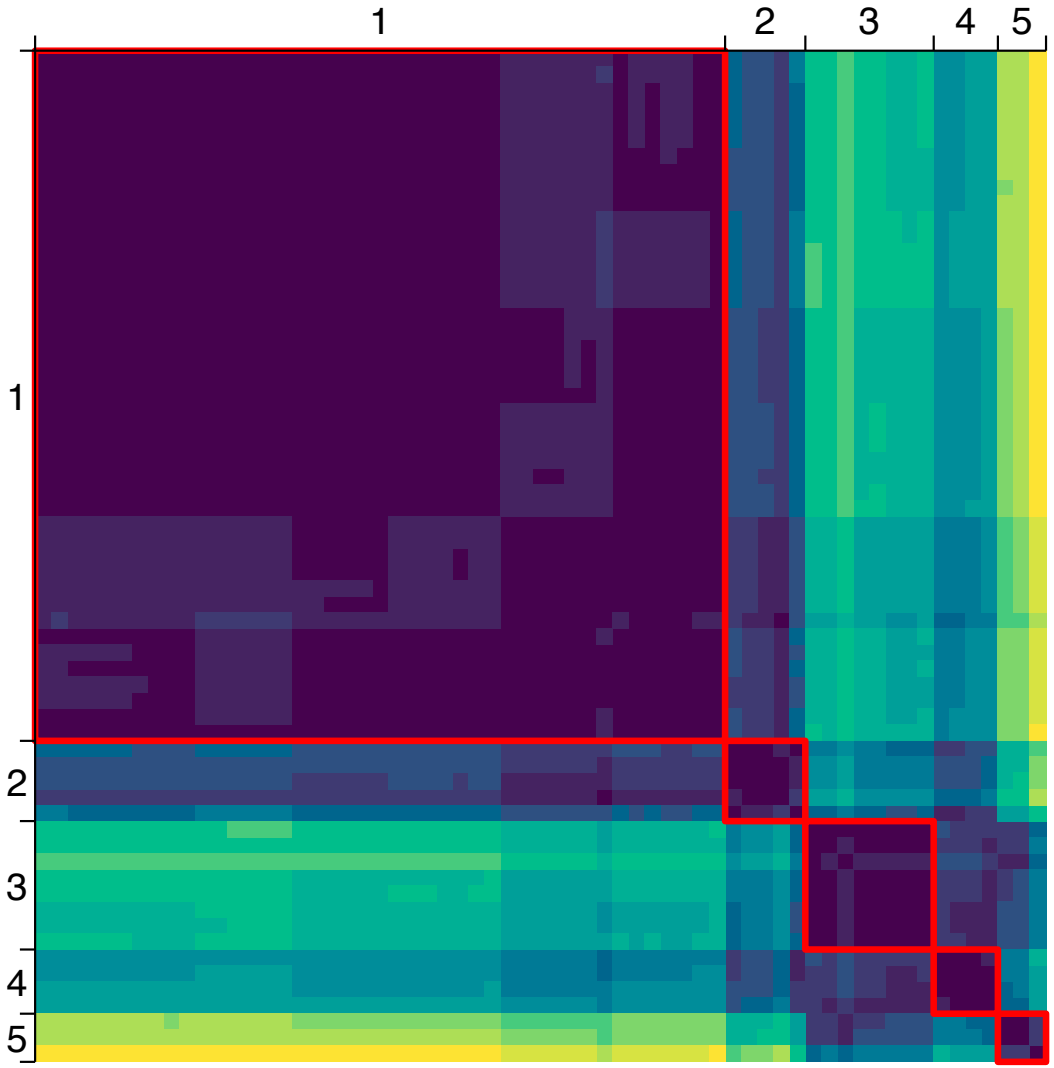


Preparation: Run a cluster analysis and obtain dissimilarity matrix; add information on clusters to the matrix

```
. sysuse lifeexp, clear
(Life expectancy, 1998)
. keep if gnppc<.
(5 observations deleted)
. cluster wards popgrowth lexp gnppc
cluster name: _clus_1
. cluster generate N = groups(`=_N'), ties(fewer)
. cluster generate G = groups(5)
. sort G N
. matrix dissim D = popgrowth lexp gnppc
. mata: st_matrixcolstripe("D", strofreal(st_data(., "G N")))
. mata: st_matrixrowstripe("D", strofreal(st_data(., "G N")))
```

Display matrix with highlighted clusters

```
. heatplot D, equations(lcolor(red) lwidth(*2)) ///
>     plotregion(margin(zero)) legend(off) aspectratio(1) xscale(alt)
```



Copy some data

```
. copy http://www.stata-press.com/data/r15/homicide1990.dta .
. copy http://www.stata-press.com/data/r15/homicide1990_shp.dta .
```

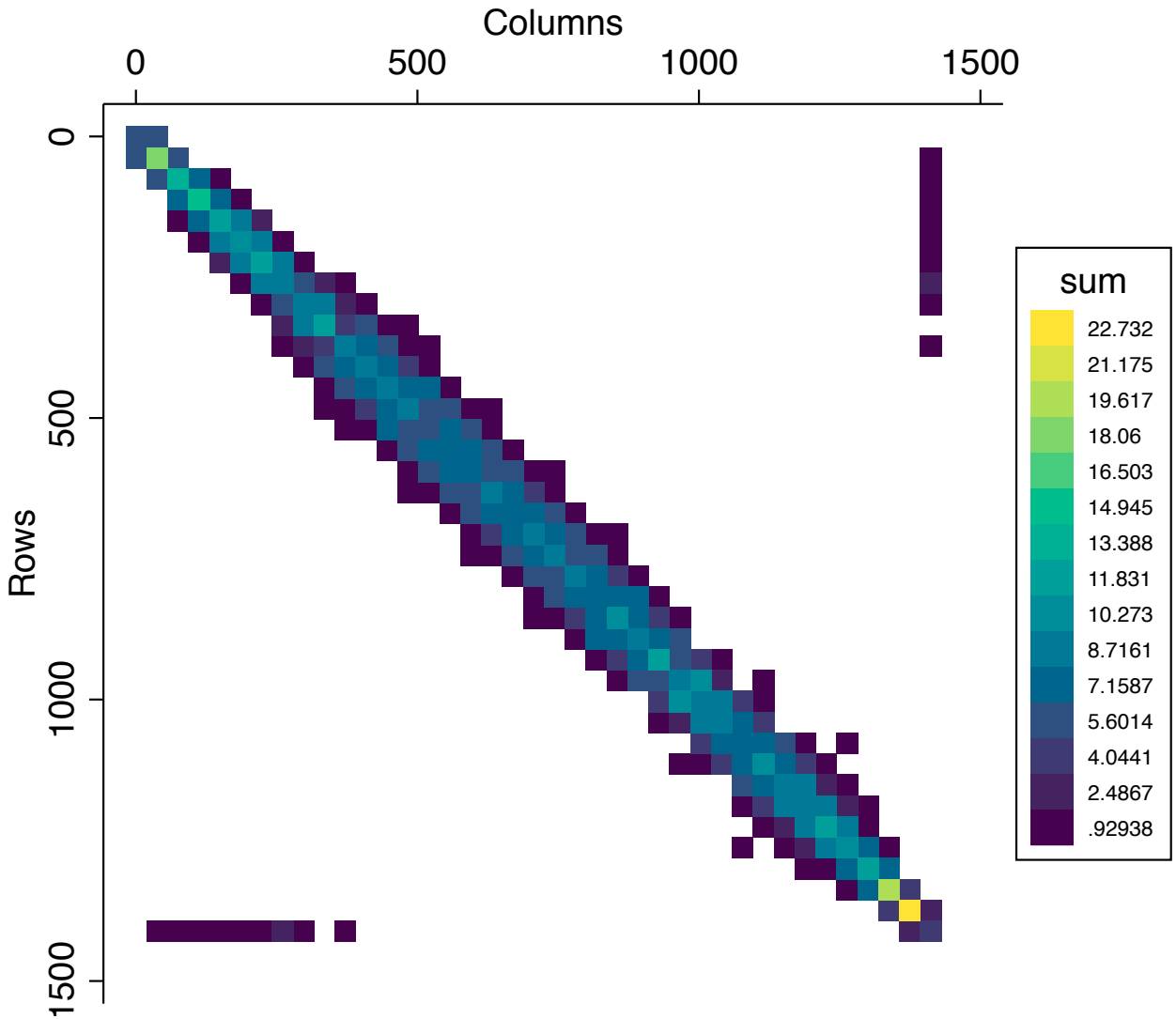
Compute spacial weights matrix (this might take a while)

```
. use homicide1990
(S.Messner et al.(2000), U.S southern county homicide rates in 1990)
. spmatrix create contiguity W
. spmatrix matafromsp W id = W
. mata mata describe W
      # bytes   type                          name and extent
-------------------------------------------------------------------------------
   15,949,952   real matrix                   W[1412,1412]
-------------------------------------------------------------------------------
```

(matrix W has about 2 million cells)

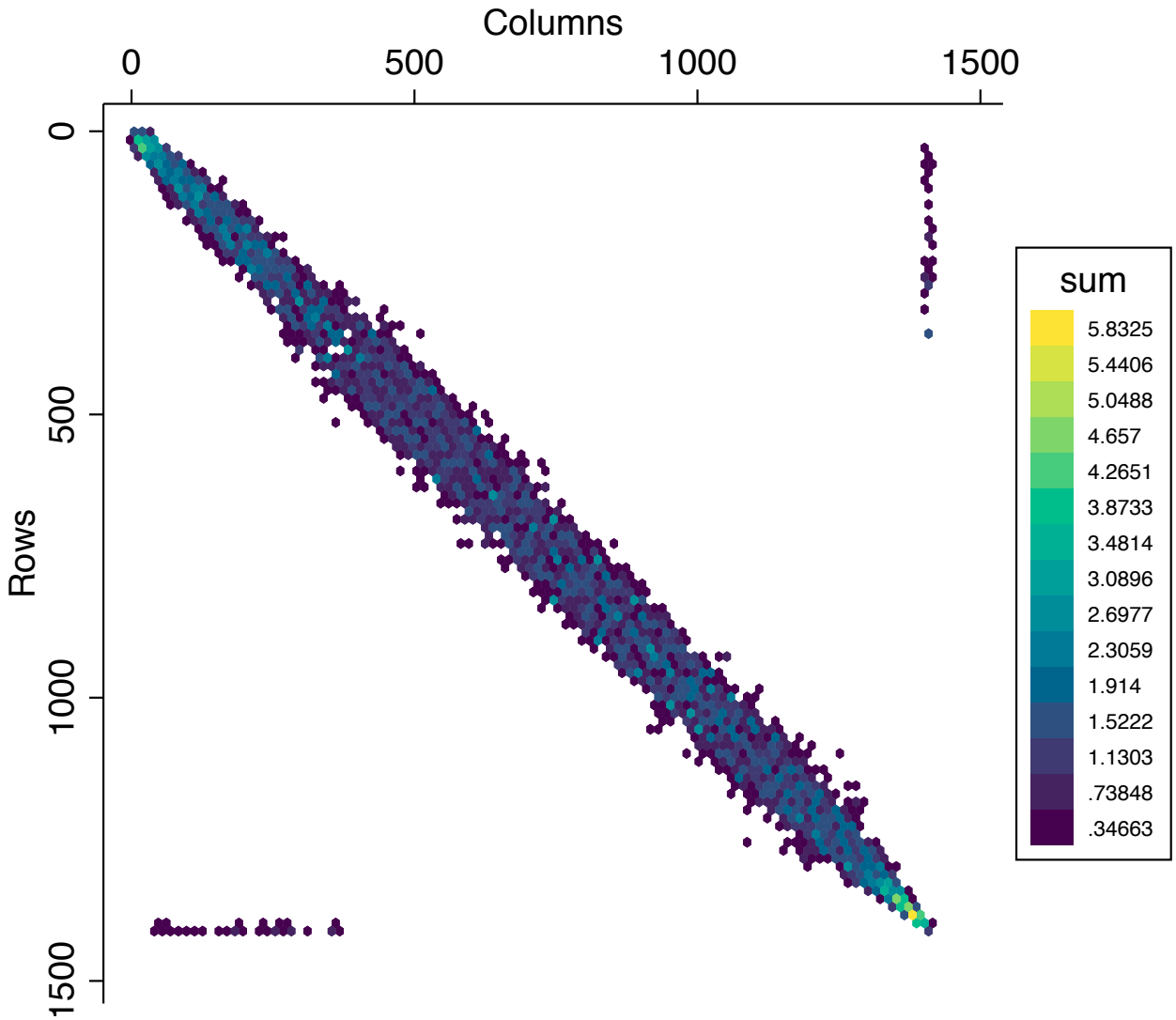
Heat plot of W with default settings, ignoring cells (i.e. weights) that are equal to zero

```
. heatplot mata(W), drop(0) aspectratio(1)
```



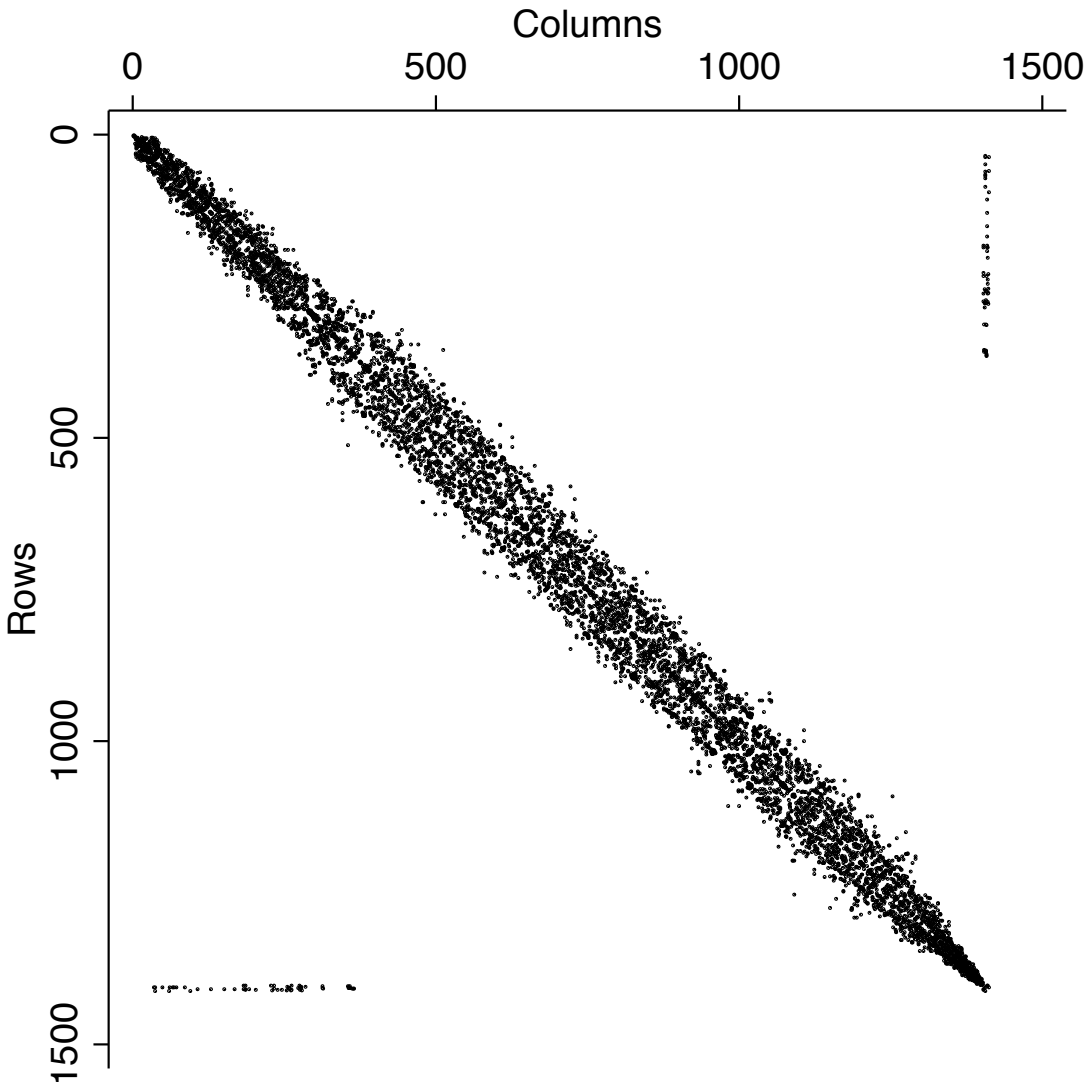
Hexagon plot with fine-grained resolution

```
. heatplot mata(W), drop(0) aspectratio(1) hexagon bins(100)
```



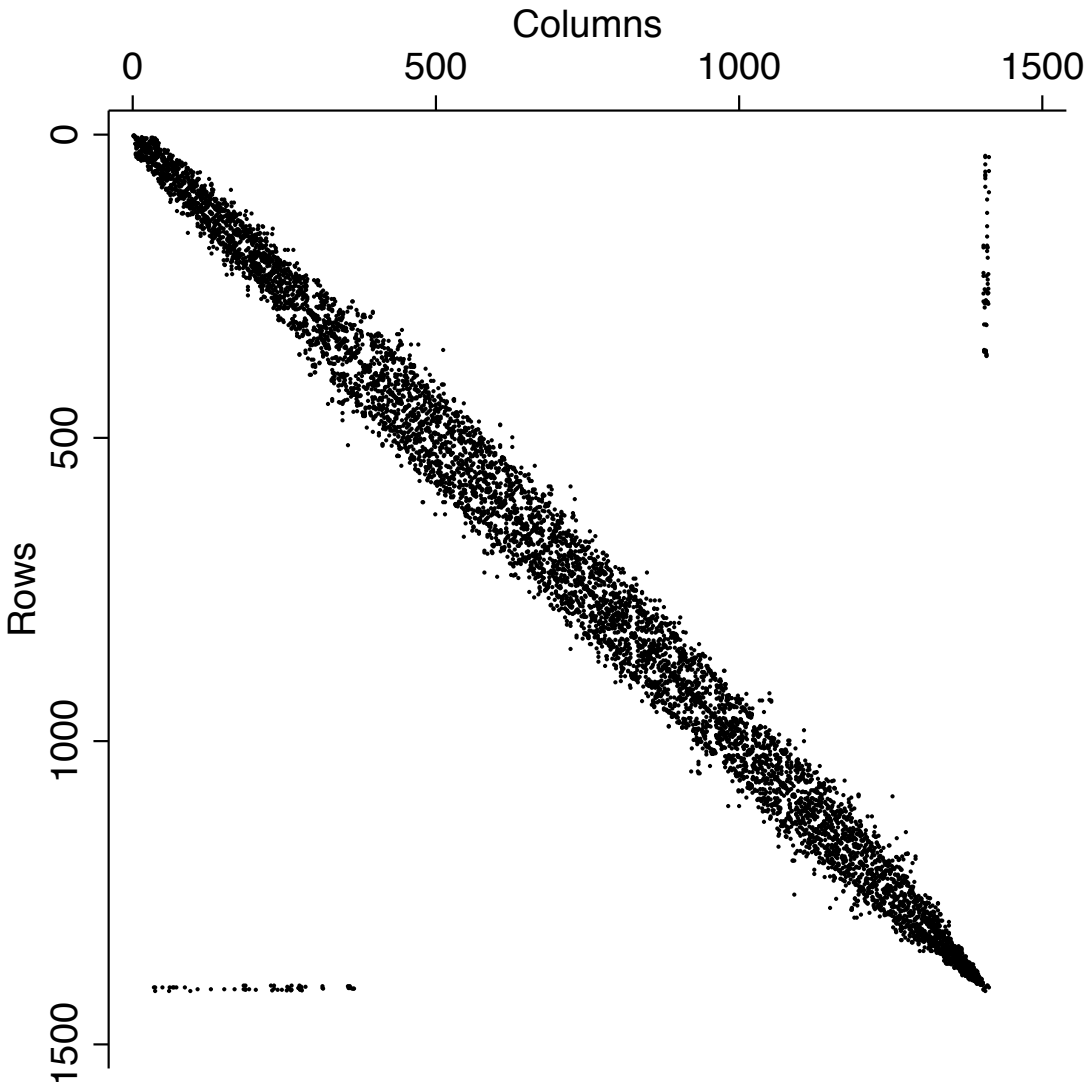
Plot each cell individually using the `discrete` option

```
. heatplot mata(W), drop(0) aspectratio(1) discrete color(black) p(lalign(center))
```



Could also use the `scatter` option

```
. heatplot mata(W), drop(0) aspectratio(1) discrete color(black) scatter p(ms(p))
```



Installation

- To install `heatplot` (and `hexplot`) type

```
. ssc install heatplot, replace
```

- `heatplot` depends on the `palettes` package, which itself depends on the `ColrSpace` Mata library, so you may also want to type

```
. ssc install palettes, replace
. ssc install colrspace, replace
```