Variable-Resolution Heat Maps

Chris Comiskey

2017-08-07

Variable-resolution (VR) heat maps integrate region-appropriate resolutions, and convey the relative data abundance of data in those regions. This vignette introduces a motivating dataset, then shows how to use `varyres()` to create a VR heat map.

You can download mapapp from GitHub with a function in the devtools package.

```
library(devtools)
devtools::install_github('cwcomiskey/mapapp')
```

Data: `hitter`

`hitter` contains 9177 rows/observations, one for each swing baseball player Jhonny Peralta took between 2008 and 2015. Each observation includes a pitch location and a swing outcome. The data is called PITCHf/x data, and comes from Sportvision, Inc in conjunction with Major League Baseball Advanced Media.

```
head(hitter)
```

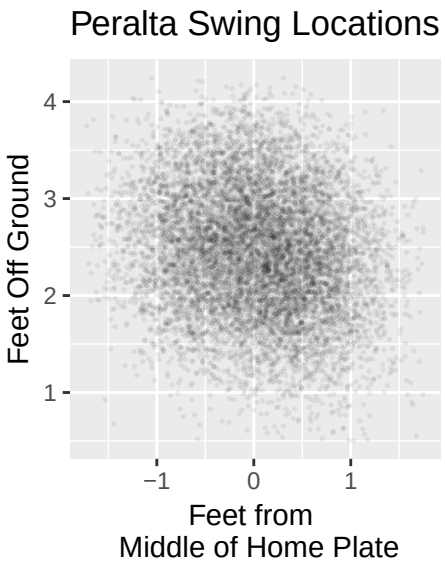
```
##        x     y res             des
## 1  0.245 2.716   0 In play, out(s)
## 2  0.257 2.096   0            Foul
## 3 -0.047 2.432   0 In play, out(s)
## 4  0.900 2.100   0 In play, out(s)
## 5  0.130 2.434   1 In play, no out
## 6  0.313 2.766   0 In play, out(s)
```

`hitter` has four columns/variables for each swing.

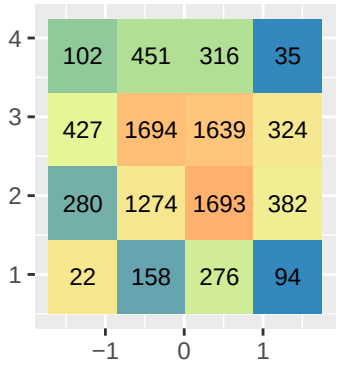
- `x` gives the horizontal location of the pitch as it passes through the strike zone, in feet from the middle of home plate.
- `y` gives the vertical location of the pitch as it passes through the strike zone, in feet from the ground.
- `res` is a Bernoulli random variable that equals 1 if the swing was successful, and 0 if not.
- `des` gives a short description of the play.

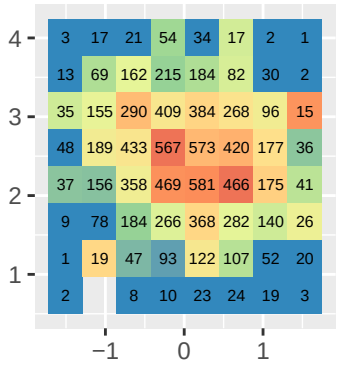
Motivation

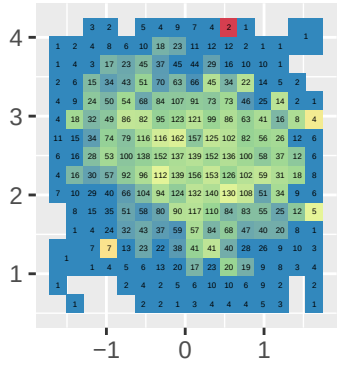
Imagine you have a continuous domain spatial dataset and you want to make a heat map. However, some regions of the domain have many more observations than others, so you are having trouble choosing the best resolution. For example, the observation dispersion might look like this.



If the whole domain looked as dense as the center we would choose one resolution; but if it looked as sparse as the edges we would choose another. Which region should primarily inform our choice? These heat maps, with box sample sizes printed, give a few possibilities.







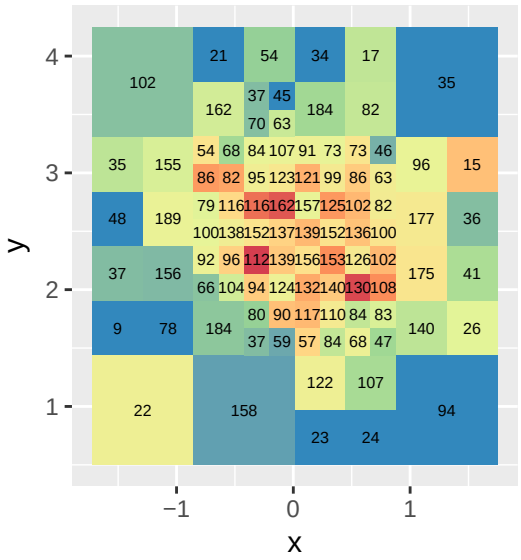
In the first map the margins look reasonable, but the central boxes contain thousands of swings each. The second map improves the graphic, but leaves some boxes virtually empty (one actually empty). The third map looks appropriate in the middle, but quite a few boxes toward the margins now have very few observations. VR heat maps integrate the resolutions appropriate for each region.

A Variable-Resolution Heat Map

The function `varyres(...)` creates, from your data set, a VR-ready data frame. From there we can use `ggplot2` to create the VR heat map.

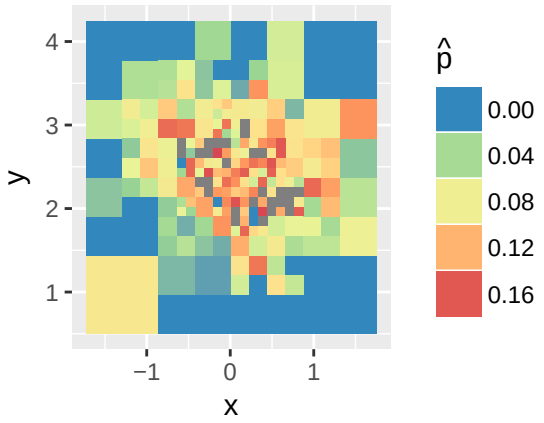
The VR algorithm works by subdividing individual grid boxes until their sample size drops below a user-specified threshold—the `cutoff` argument.

```
vr <- varyres(data = hitter, cutoff = 200, max = 4)
```



The VR map has finer resolution in the center, more coarse resolution around the edges, and in-between as needed. Also, notice how the box sizes implicitly convey the varying data concentration: bigger boxes correspond to less data, smaller boxes to more data. The sample sizes printed on the grid boxes explicitly show this correspondence.

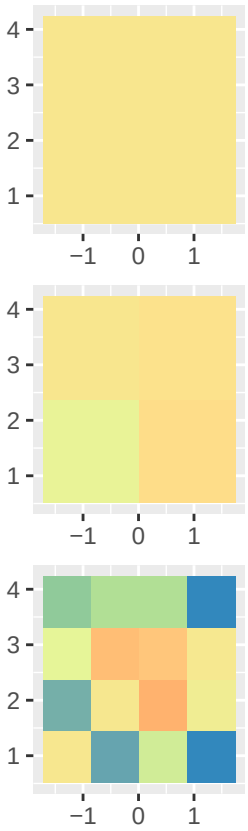
The `cutoff` argument gives the user control over how populated boxes will be. For example, `cutoff = 100` gives this map.

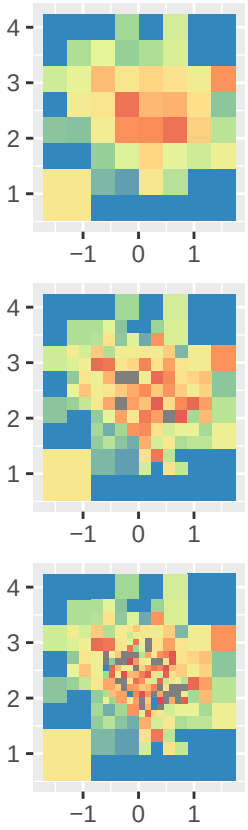


Notice one prominent difference: smaller central boxes, due to further subdivision.

Iterations

`varyres(...)` returns the subdivision iterations, and seeing these steps helps understand how the algorithm works. The sequence of subdivisions here proceeds by subdividing boxes with sample sizes above `cutoff = 100`.




4
3
2
1
−1
0
1
4
3
2
1
−1
0
1
4
3
2
1
−1
0
1