datplot - Density Plots for Dates

Lisa Steinmann

28 Mai 2018

Density Plots for Dates

A rather common problem in archaeology is the fuzziness of dates assigned to objects. If one wants to visualize overall changes in - let's say - pottery consumption, bar charts often fall short in that regard. If we have Phases a − f, then some of the objects can usually be dated to a, c, and f, as an example, but others will by classified as "a to c" or "b to c". But how can these data still be used for examining changes in a large set of objects?

First, it is handy to translate the phases into numbers, for which we should conveniently choose the 'actual' dating. This may cause other problems in the end, since such phases are often employed to avoid dates, but it is necessary as the aim is to visualize the distribution on a continuous scale, for which numbers are needed. Also, this step may be reversed for the final visualization but supplementing or replacing the scale on the x-axis with the respective phases. Ideally, one can produce a 'beginning' and 'end' date for each object, or let's say an earliest possible dating and a latest possible dating, e.g. corresponding to beginning and start of each phase the object is dated to.

To show and explain how this would work, I chose a random sample of athenian pottery from the beazley archive (("Beazley Archive Pottery Database (Bapd)," n.d.)), as it is a large publicly available dataset. (Since the format provided by the BAPD is slightly different from that needed here I converted the data beforehand to match my requirements. No values have been changed.)

```r
df <- read.table("../inst/data/testset_beazley_1000.csv", sep = ";")
kable(df[sample(1:nrow(df), 10, replace = FALSE),])
```

	Vase.Number	Technique	DAT_min	DAT_max
32730	46938	BLACK-FIGURE	-550	-500
14401	15602	BLACK-FIGURE	-550	-500
11338	12355	RED-FIGURE	-425	-375
39013	206179	RED-FIGURE	-500	-450
53751	275925	RED-FIGURE	-500	-450
12419	13495	BLACK-FIGURE	-525	-475
24236	28398	BLACK-FIGURE	-525	-475
65129	351824	BLACK-FIGURE	-525	-475
23101	25439	RED-FIGURE	-450	-400
30119	43775	RED-FIGURE	-425	-375

How to display a range?

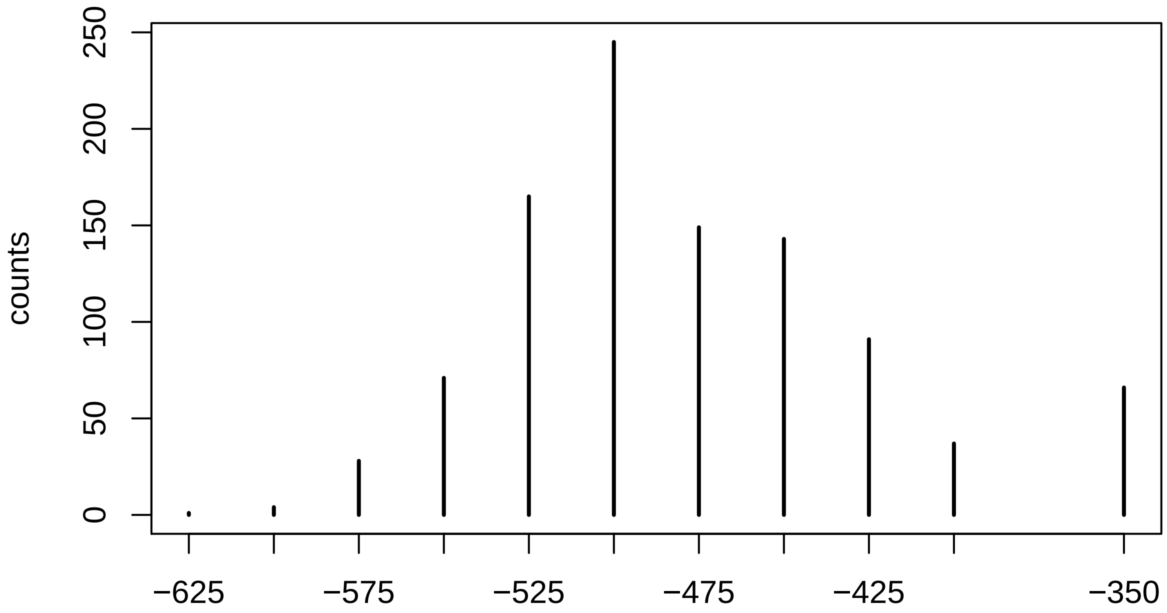
The table provides two dates for each object. The earliest possible dating (DAT_min) and the latest possible dating (DAT_max). In order to be able to process this to a density graph, which is the most elegant means of visualization for continuous distributions. (At least if the goal is merely to evaluate changes over time and not to look at actual objects counts, which will be omitted.)

Objects that can be dated with greater confidence should have a larger impact on the overall visualisation. The core function of this package (`datsteps()`) produces a column named 'weight' which contains a value that corresponds to one (as the closest possible dating of one year) devided by the timespan between the two dating variables. The greater the timespan, the lower the weight value. Secondly, every object is duplicated a number of times equal to the dating range devided by the stepsize-variable. Each duplicate has its own 'date', one single value between the two extremes. The above mention weight variable is devided by the number of steps, so that each new fictional object or 'date step' counts only as a fraction of the actual object.

This method will not be useful for dating specific context, since any concept of *terminus post/ante quem* is lost here, which is important on a smaller scale. For the visualization of changes in *trends* over time, e.g. the popularity of pottery types, or overall peaks in occupation from survey data, the method is ideal.

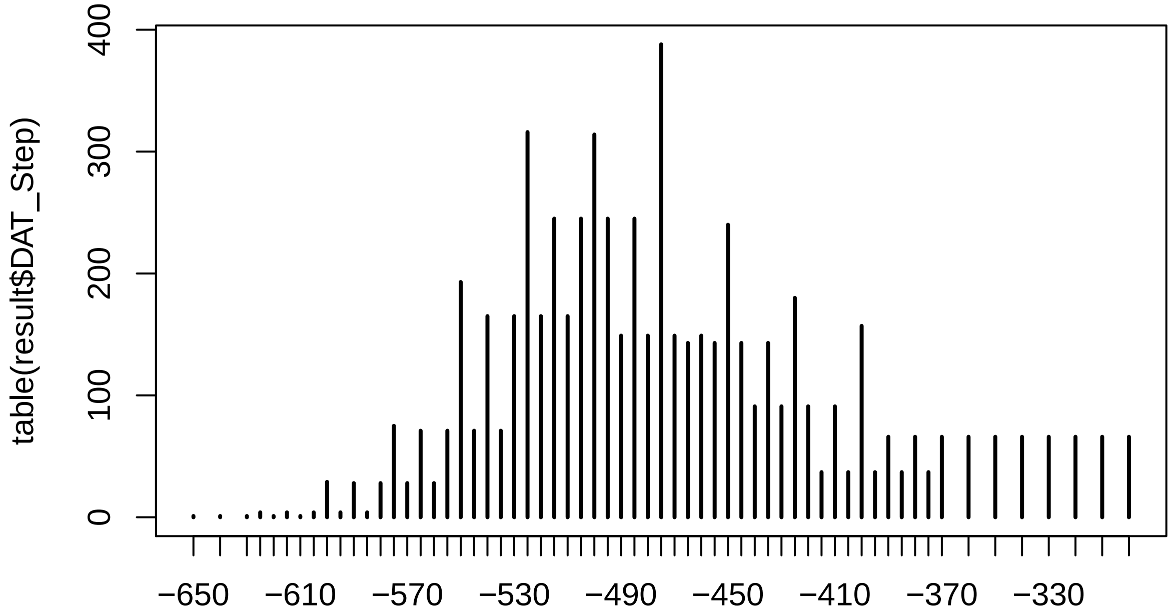
Other approaches, e.g. using the median date of each object, may often produce similar outcomes, but create other problems. A lot of information is lost on the way when employing averaged or median data, as large amount of loosely dated objects will produce peaks at unreasonable values. (Consider a large amount of objects dated between 600 and 400 BCE all attributed to the year 500 BCE.)

```r
counts <- df
counts$DAT_med <- ((counts$DAT_max + counts$DAT_min) / 2)
counts <- table(counts$DAT_med)
plot(counts)
```



Employing dating steps will even out unreasonable peaks. Not especially the gap between -425 and -300 in the plot above, that is – in the plot below – filled with a constant amount of objects in each year. This is due to the data containing large amounts of objects dating from -400 to -300 BCE. Of couse, due to duplicating each object numerous times (see table below), the counts represented on the y-axis have become meaningless.

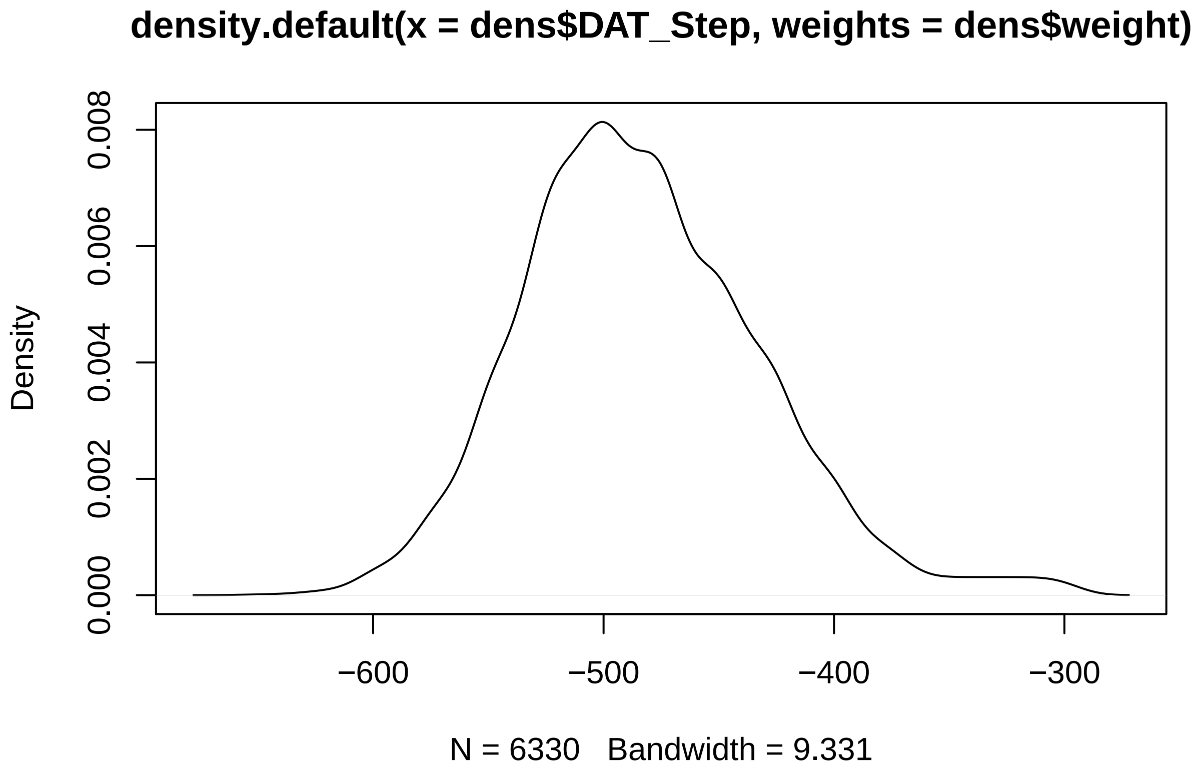
```r
library(datplot)
result <- datsteps(df, stepsize = 10)
plot(table(result$DAT_Step))
```



	Vase.Number	Technique	DAT_min	DAT_max	weight	DAT_Step
10110	10957	BLACK-FIGURE	-550	-500	0.0033333	-550
101101	10957	BLACK-FIGURE	-550	-500	0.0033333	-540
101102	10957	BLACK-FIGURE	-550	-500	0.0033333	-530
101103	10957	BLACK-FIGURE	-550	-500	0.0033333	-520
101104	10957	BLACK-FIGURE	-550	-500	0.0033333	-510
101105	10957	BLACK-FIGURE	-550	-500	0.0033333	-500

Due to the impossibility of displaying object counts as well, it is ideal to use kernel density estimates for visualization. The density plot below shows the result. The peak at around -500 indicates that is area has the highest overlay, so a large part of the objects in our sample have been dated around this time. The same distribution can also be seen in the bar plots above. This, however, is not yet very informative.

```
dens <- result
dens$weight <- (dens$weight / sum(dens$weight))
dens <- density(x = dens$DAT_Step, weights = dens$weight)
plot(dens)
```



Scaling the weight along groups of objects

In order to display the objects seperated into groups, the weights first have to be scaled along group membership, so that the sum of all weights in a group will equal 1. datplots function `scaleweight()` does exactly that for a dataframe as it was returned by `datsteps()`. A column that contains the variables for group membership needs to indicated.

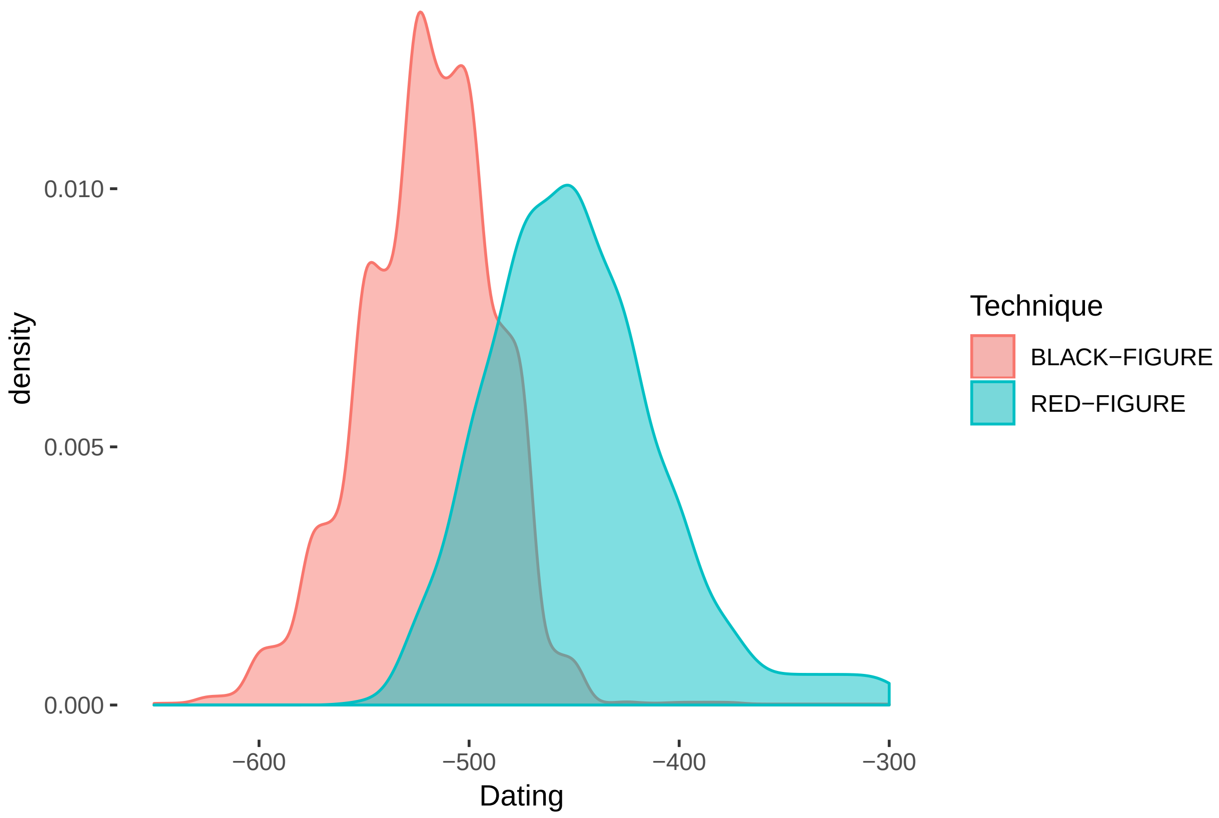
```
result <- scaleweight(result, result$Technique)
kable(head(result))
```

	Vase.Number	Technique	DAT_min	DAT_max	weight	DAT_Step
10110	10957	BLACK-FIGURE	-550	-500	0.0003501	-550
101101	10957	BLACK-FIGURE	-550	-500	0.0003501	-540
101102	10957	BLACK-FIGURE	-550	-500	0.0003501	-530
101103	10957	BLACK-FIGURE	-550	-500	0.0003501	-520
101104	10957	BLACK-FIGURE	-550	-500	0.0003501	-510
101105	10957	BLACK-FIGURE	-550	-500	0.0003501	-500

Plots for the distribution of objects across time

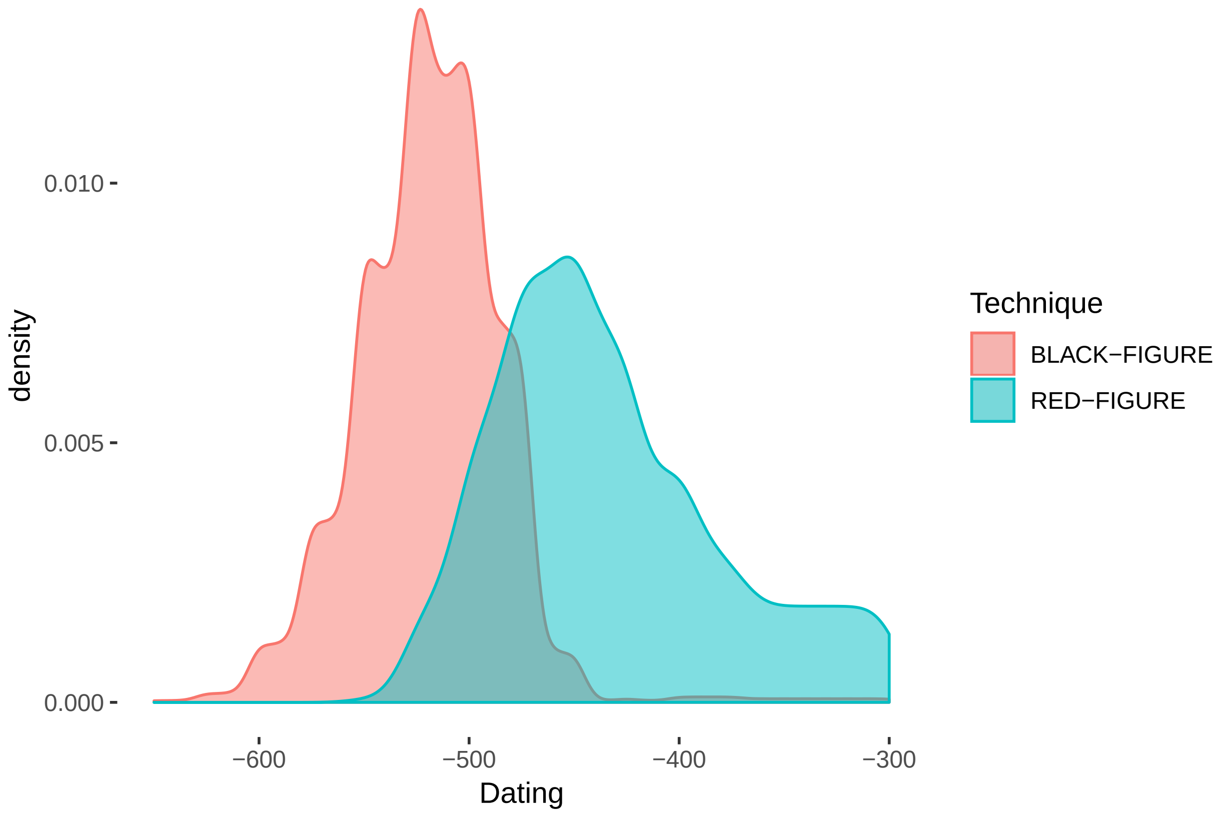
In the case of the beazley archives data ("Beazley Archive Pottery Database (Bapd)," n.d.) we can clearly see what we knew before: Black-figure pottery is older than red-figure pottery. (The data are from a random sample of athenian pottery from the beazley archive, n = 1000. Computation of the dating steps may not work with very, very large datasets, or simply take up a lot of time.)

```
library(ggplot2)
ggplot(data = result, aes(x = DAT_Step,
                          color = Technique,
                          fill = Technique)) +
  geom_density(aes(weight = weight), alpha = 0.5) +
  xlab("Dating") +
  theme(panel.background = element_blank())
```



Please note that the plot does change when the weights are omitted (see plot below). When every step is valued equally, a lot of steps fall into the end of the 4th century (as mentioned above), since they were dated as e.g. “-400 to -300”.

```
ggplot(data = result, aes(x = DAT_Step,
                          color = Technique,
                          fill = Technique)) +
  geom_density(alpha = 0.5) +
  xlab("Dating") +
  theme(panel.background = element_blank())
```



The smooth curves of kernel density estimates are a more realistic approach to dating. The production of objects was as continuous as their use, so it seems only reasonable to display it in a continuous fashion.

References

“Beazley Archive Pottery Database (Bapd).” n.d. https://www.beazley.ox.ac.uk/pottery/default.htm.