

# datplot - Density Plots for Dates

*Lisa Steinmann*

*28 Mai 2018*

## Density Plots for Dates

A rather common problem in archaeology is the fuzzyness 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. Let's say 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 be classified as “a to c” or “b to c”. But one 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. I know this causes other problems in the end, since such phases are often employed to avoid dates, but it is necessary in this case and may as well be changed back later in the visualization. Ideally, we can produce a ‘beginning’ and ‘end’ date for each object, or let's say an earliest possible dating and a latest possible dating corresponding to beginning and start of each phase mentioned above.

I will demonstrate an approach of visualization that I personally found very useful, although I guess it is debatable whether it's ‘valid’ or not - I find that it works, however, and am very open to discussion. Let's start with a random sample of athenian pottery from the beazley archive ((“Beazley Archive Pottery Database (Bapd),” n.d.)), that I converted into the format described above beforehand!

```
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
98858	9019787	RED-FIGURE	-500	-450
9307	10114	BLACK-FIGURE	-575	-525
57250	303311	BLACK-FIGURE	-550	-500
63076	340188	RED-FIGURE	-500	-450
64311	350956	BLACK-FIGURE	-550	-500
47463	215346	RED-FIGURE	-450	-400
23925	28072	RED-FIGURE	-475	-425
17776	19547	BLACK-FIGURE	-550	-500
36680	203822	RED-FIGURE	-500	-450
26783	31047	BLACK-FIGURE	-525	-475

## How to display a range?

Now we have to dates for each object. The earliest possible dating and the latest possible dating. In order to be able to process this to a density graph, which to me seems the most elegant means of visualization for this kind of data. (At least if the goal is merely to evaluate changes over time and not to look at actual objects counts.)

In my opinion, things that can be dated with greater confidence should have a larger impact on the overall visualisation. That is why the following function produces a column named ‘weight’ which contains a value that corresponds to one (as the closest possible dating of one year) divided by the timespan between by 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 divided by the stepsize-variable. Each duplicate has its own ‘date’, one single value between the two extremes. The above mention weight variable is divided

by the number of steps, so that each new fictional object or ‘date step’ counts only as a fraction of the actual object. In this case the size of the dating steps will be 10 years. I find this superior to using, e.g., the median date of an objects. The outcome might often be similar, but I still have the impression that a lot of information is lost on the way when doing that. I do not claim this to be an ideal solution to everything, however. There *are* still problems and it might not be suitable for every purpose. I can’t imagine this being useful for ‘fine’-dating context, since it does erase the point of terminus post/ante quem, which is important on a smaller scale. I would rather imagine it being usefull to display the change in ‘trends’ over time, e.g. which vessel type is popular at a given time? How does style develop? Or maybe even to find occupation ‘peaks’ from survey data?

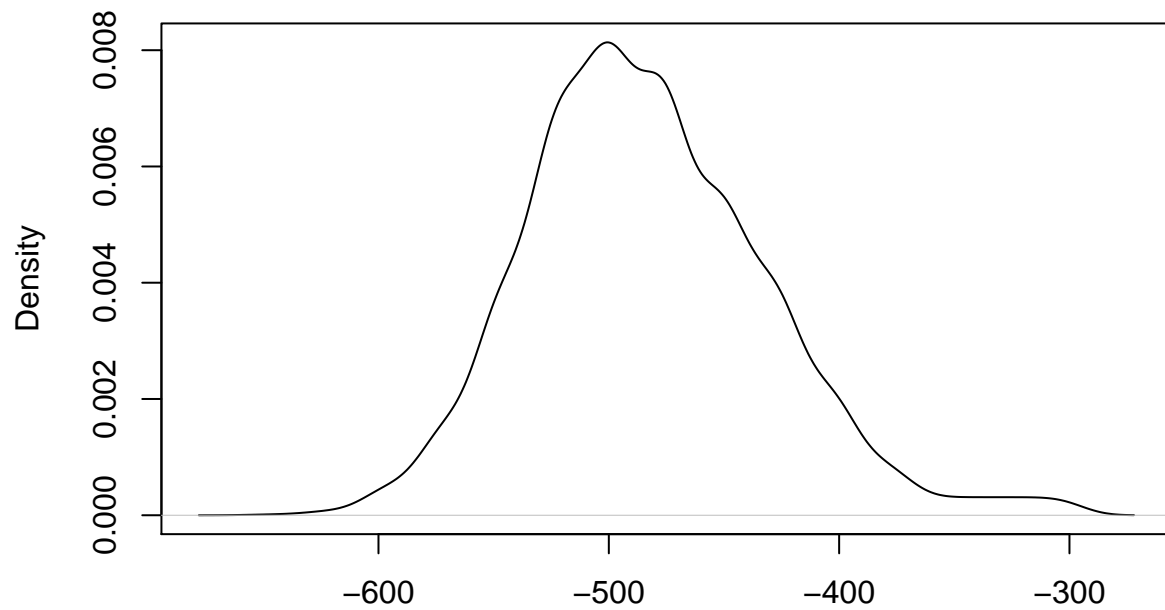
```
library(datplot)
result <- datsteps(df, stepsize = 10)
kable(head(result))
```

	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

This can now be displayed as a density plot, as seen below. 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. 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)
```

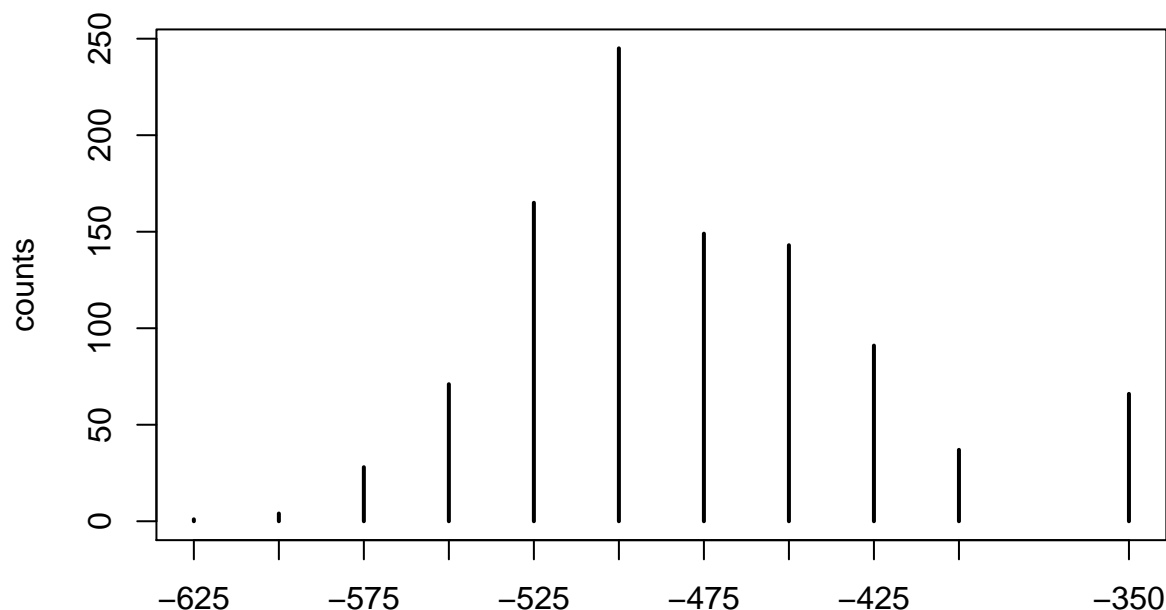
**density.default(x = dens\$DAT\_Step, weights = dens\$weight)**



N = 6330 Bandwidth = 9.331

A simplistic approach to a bar plot however would look like this:

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



## Scaling the weight along groups of objects

But: In order to display the objects separated 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. `scaleweight()` does exactly that for a dataframe as it was returned by `datsteps()`. We also need to supply the group column.

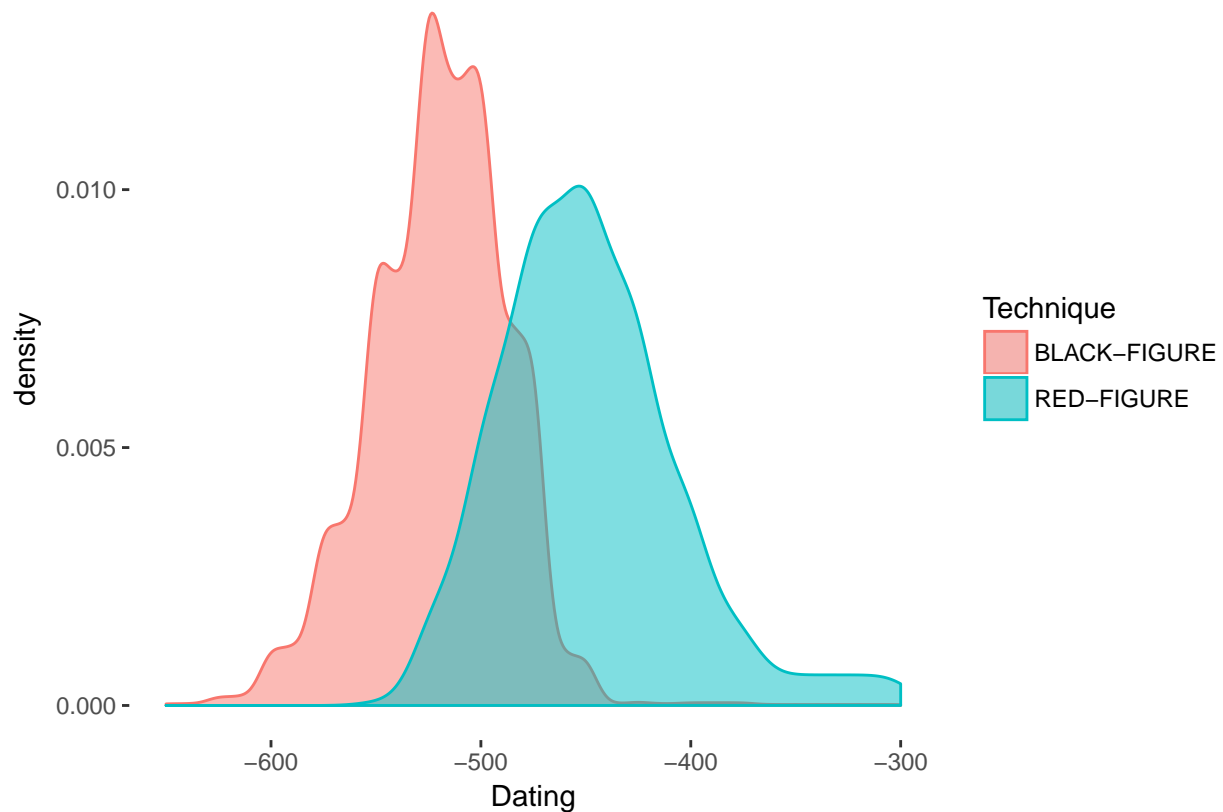
```
result <- scaleweight(result, result$Technique)
kable(head(result))
```

	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

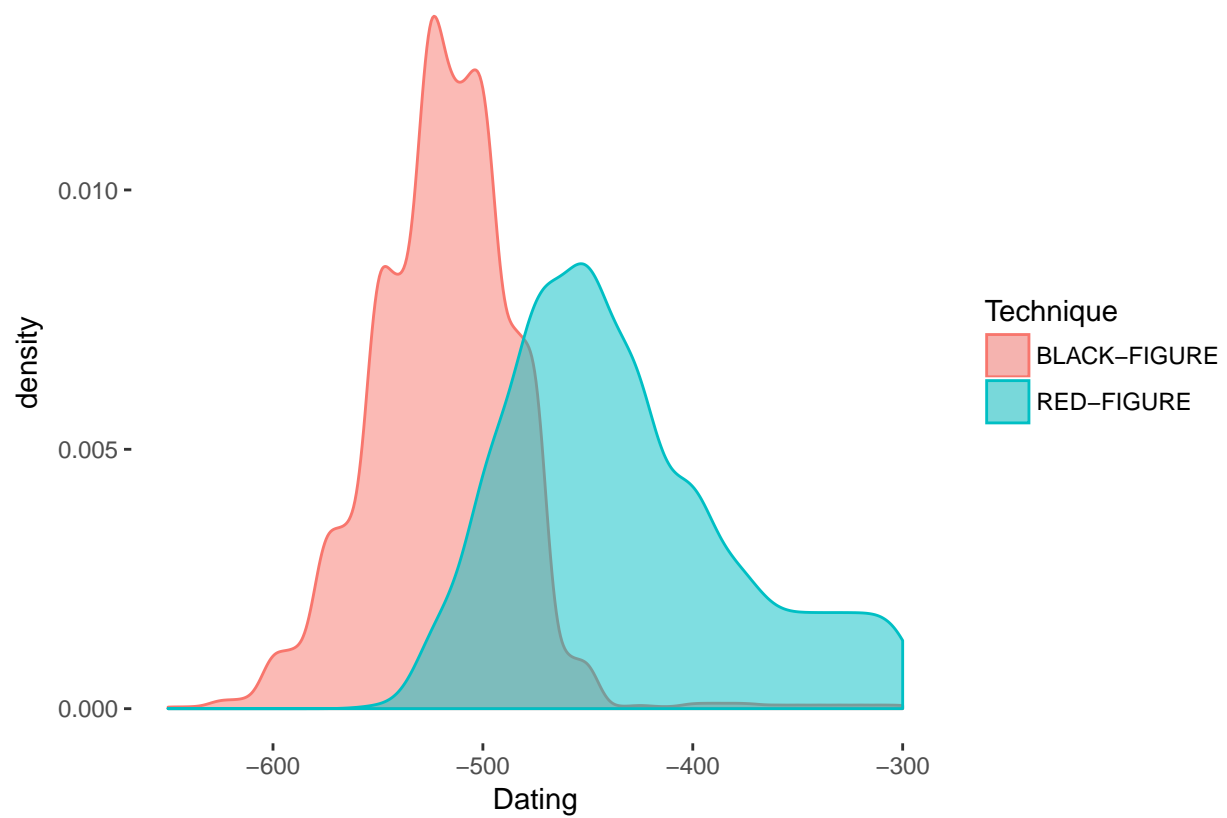
This can now be plotted a little more nicely using `ggplot2`. We can clearly see now what we knew before: Black-figure pottery is older than red-figure pottery. (The data are from a random sample of athenian pottery from the beazley archive,  $n = 1000$ . Computation of the dating steps may not work 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())
```



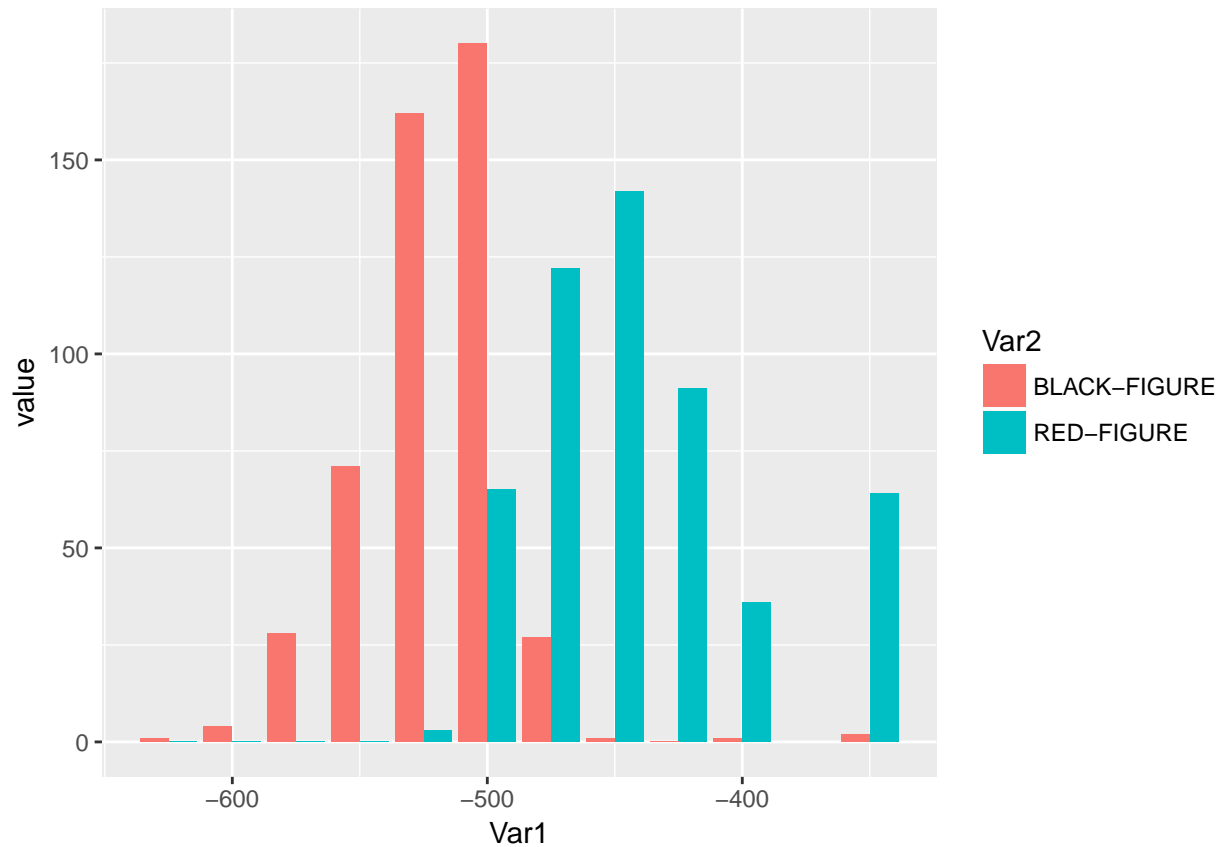
Please note that the plot does change when the weights are omitted. When every step is valued equally, a lot of steps fall into the end of the 4th century, since they were dated as e.g. “-400 to -300”.

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



### Barplots for comparison

```
counts <- df
counts$DAT_med <- ((counts$DAT_max + counts$DAT_min) / 2)
counts <- table(counts$DAT_med, counts$Technique)
library(reshape2)
counts <- melt(counts)
ggplot(data = counts, aes(x = Var1, y = value, fill = Var2)) +
  geom_bar(stat = "identity", position = "dodge")
```



So, this is similar, but not quite. I find the smooth curves a more realistic approach to dating, since nothing should be fixed at one certain year alone. The production of objects was as continuous as there 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>.