Inscriptions of Bithynia: A Case Study on Nicaea, and Workflow with datplot v.1.0.0

Supplementary Material to 'DATPLOT: A NEW R-PACKAGE FOR THE VISUALIZATION OF DATE RANGES IN ARCHAEOLOGY'

Barbora Weissova and Lisa Steinmann

Contents

1 The Data Set

The data used in this supplement was compiled for a publication on the economic development of ancient Bithynia (Weissova 2019). A large amount of the initial data was provided by the Searchable Greek Inscriptions Tool of the Packard Humanities Institute (https://inscriptions.packhum.org/) and the Epigraphische Datenbank Heidelberg (https://edh-www.adw.uni-heidelberg.de/home). Those inscriptions (n = 2545) can be referenced in the databases with their `ikey`, which are prefixed with `HD` for the latter and `PH` for the former. For convenience, the respective URLs are attached to the data in a separate column. The assemblage was supplemented with 333 inscriptions that were manually gathered from different sources and subsequently do not have a URL. Relevant citations for each inscription are to be found in the `Source` column. Additional information on the creation and curation of this data set can be found in the publication mentioned above (Weissova 2019) .

2 Data Preparation, Cleaning and Reformatting

As the original data set was not not suited for analysis, we had to clean and reformat it. This process is detailed in a vignette added to the `datplot` package (after installing: `browseVignettes("datplot")`- "Data Preparation and Visualization"). The final data set is available together with datplot (`data("Inscr_Bithynia")`), and all the raw data are provided in the datplot-repository as well (<https://github.com/lsteinmann/datplot> / «https://doi.org/>10.5281/zenodo.4285911>).

datplot (this supplement uses version 1.0.0) is available on CRAN and can be installed with the following command:

```
install.packages("datplot")
```

As the location and the chronology represent crucial information for the query, only located and dated inscriptions are suitable for the analysis. We therefore filter the data accordingly and include only inscriptions that have known find spot and dating:

```
library(datplot)
library(tidyverse)
data("Inscr_Bithynia")
inscr_clean <- Inscr_Bithynia %>%
  filter(!is.na(DAT_min), !is.na(DAT_max),
         Location != "unknown") %>%
  droplevels()
```

3 Preparation for the Plots

As we will repeatedly use the same plot styles and colors, we can define a theme-object for use with ggplot() and a color palette beforehand, so that the code chunks will be less cluttered:

```
bluegreen <- colorRampPalette(c("#8dae25", "#17365c"))
library(ggplot2)
Plot_Theme <- theme(legend.position = "top",
                    panel.background = element_blank(),
                    panel.grid.major = element_line("black", size = 0.5,
                                                    linetype = "dashed"),
                    panel.grid.minor = element_line("grey", size = 0.5,
                                                    linetype = "dotted"))
No_Y <- theme(axis.text.y = element_blank(), axis.ticks.y = element_blank())
Plot_Scale <-  scale_x_continuous(breaks = seq(from = -500, to = 900, by = 100),
                     limits = c(-500, 900), name = "")
```

4 Nicaea: A Case Study

Nicaea (modern Iznik) belonged to one of the most powerful ancient cities in Bithynia. It is situated on the eastern shore of the Iznik Lake, surrounded with fertile flatland and protected by mountainous ridges (Weissova et. al. 2019). The inscriptions from the territory of Nicaea were chosen for a detailed examination of the possibilities of their visualization implementing the **datplot** package out of two main reasons. Not only that the territory revealed the most numerous assemblage of dated inscriptions in Bithynia, represented by 604 out of 1498 in total, but the precision of the chronology varies between one exact year and the time-span of 426 years (the Roman Imperial Period) for individual inscriptions, allowing us to fully examine and question the functionality of the package.

The 604 objects of Nicaea thus serve as our case study to demonstrate the functionality of **datplot** as well as its workflow in this supplement. We briefly compare the output to the method employed in the original publication as well (Weissova 2019), to highlight differences and advantages.

First, the data set is selected:

```
inscr_Nicaea <- inscr_clean %>%
  filter(Location == "Nicaea") %>%
  select(ID, Location, Source, URL, Dating, Language, DAT_min, DAT_max)
```

A random sample can serve as an impression of the data now available to us. We exclude the URL to save space:

```
URL <- which(colnames(inscr_Nicaea) == "URL")
inscr_Nicaea[sample(nrow(inscr_Nicaea), size = 5),-URL]
```

ID	Location	Source	Dating	Language	DAT_min	DAT_max
I_1442	Nicaea	IK Iznik 1055	1st – 2nd c. AD	Greek	1	199
I_916	Nicaea	IK Iznik 1007	Roman Imp. period	Greek/Latin	-31	395
I_986	Nicaea	IK Iznik 1595	Roman Imp. period	Greek	-31	395
I_1356	Nicaea	IK Iznik 1328	2nd c. AD	Greek	100	199
I_1456	Nicaea	IK Iznik 1483	1st – 2nd c. AD	Greek	1	199

4.1 datsteps() and Time

The `datsteps()` function can automatically determine the lowest stepsize actually occurring in the data, which in our case should be 1, as the assemblage encompasses inscriptions dated to a single year. Using datplot with a suitable data set is easy. We select the columns we need in the order that has to be handed to `datsteps()` (ID, variable, minimum date, maximum date), and let `datsteps()` decide what stepsize to use:

```
library(datplot)
Nicaea_steps <- inscr_Nicaea %>%
  select(ID, Language, DAT_min, DAT_max)
```

```
datsteps(Nicaea_steps, stepsize = "auto")
```

```
## [1] "Using stepsize = 1 (auto)."
```

```
## Warning in get.weights(DAT_mat[, "datmin"], DAT_mat[, "datmax"]): Warning:
## DAT_min and DAT_max at Index: 284, 525, 526, 530, 536, 579, 584, 587, 588, 590,
## 591, 592, 599) have the same value! Is this correct? Please check the table for
## possible errors.
```

The function then returns a warning which notifies us of the inscriptions which are dated to a single year. We can check them using the index values (row numbers in the `data.frame`) provided by datplot to verify that this is indeed correct:

```
inscr_Nicaea %>%
  select(ID, Dating, DAT_min, DAT_max, URL) %>%
  slice(c(284, 525, 526, 530, 536, 579, 584, 587, 588, 590, 591, 592, 599))
```

ID	Dating	DAT_min	DAT_max	URL
I_1165	58 AD	58	58	https://epigraphy.packhum.org/text/277805
I_1406	269 AD	269	269	https://epigraphy.packhum.org/text/277803
I_1407	269 AD	269	269	https://epigraphy.packhum.org/text/277804
I_1411	244 AD	244	244	https://epigraphy.packhum.org/text/278114
I_1417	210 AD	210	210	https://epigraphy.packhum.org/text/278458
I_1460	199 AD	199	199	https://epigraphy.packhum.org/text/278107
I_1465	146 AD	146	146	https://epigraphy.packhum.org/text/278207
I_1468	131 AD	131	131	https://epigraphy.packhum.org/text/278191
I_1469	128 AD	128	128	NA
I_1471	123 AD	123	123	https://epigraphy.packhum.org/text/277814
I_1472	123 AD	123	123	https://epigraphy.packhum.org/text/277815
I_1473	123 AD	123	123	https://epigraphy.packhum.org/text/277816
I_1480	116 AD	116	116	https://epigraphy.packhum.org/text/278101

We implemented this as a warning, as such precise chronological assessments are rare and might be an accidental result of the process of data cleaning in some cases. As they greatly influence the resulting distribution, it seemed reasonable to routinely check for these occurrences. Using the URL provided in the data set, it is possible to conveniently take a closer look at the inscriptions in question and verify their dating, which is indeed identical with the available database entries. Specific archaeological questions might arise at this point, which could pertain the reasons for such precise dating and the events and historical figures associated with the inscriptions (Weissova 2019:130–150).

To illustrate that the process can be a bit time consuming, we recorded the time it took to process `datsteps()` with the stepsize of 25 vs. the automatically determined stepsize of 1. The difference might not seem huge now, but for larger data sets this can be more relevant.

```
system.time(datsteps(Nicaea_steps, stepsize = 25))
```

```
##    user  system elapsed
##    0.28    0.01    0.31
```

```
system.time(datsteps(Nicaea_steps, stepsize = "auto"))
```

```
## [1] "Using stepsize = 1 (auto)."
```

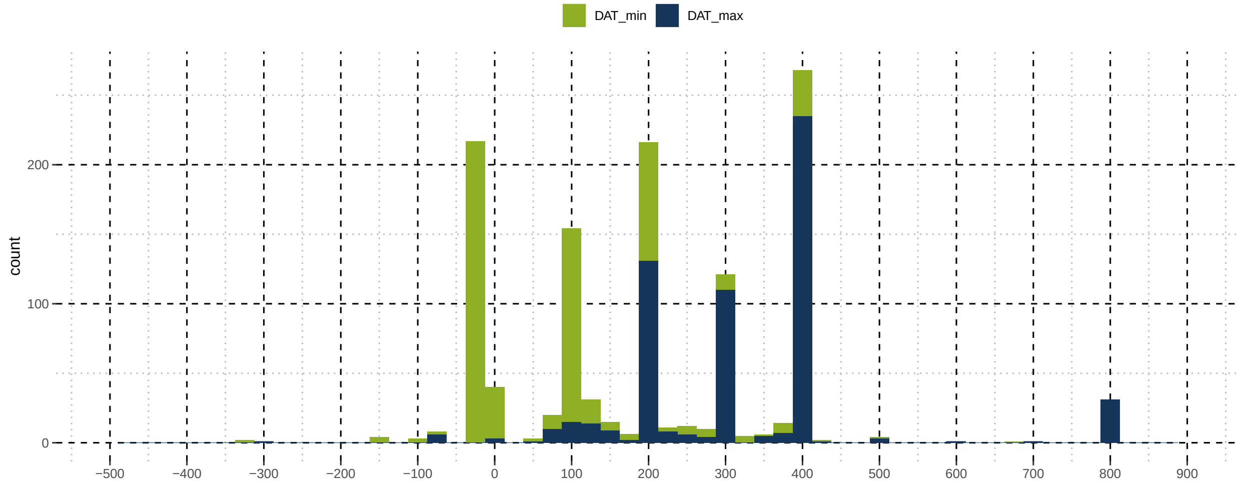
```
##    user  system elapsed
##    3.77    2.96    6.87
```

As you can see, a stepsize of 25 takes only 0.31 seconds to compute, while using the stepsize of 1 amounts to an elapsed time of 6.87 seconds – of course, depending on the machine one uses. This difference grows exponentially with larger `data.frame`s. In previous updates, we have streamlined the process in order to save some time, but there still seems to be a lot of room for improvement. We retained the functionality in the package, as we think it might be helpful in some cases, i.e. for truly large data sets. It may also be convenient to speed up the process for researchers looking at far greater timespans, where changes within even a single century would hardly matter. We warmly invite anyone interested to fork and change the code of **datplot** we provide on GitHub (<https://github.com/lsteinmann/datplot> / <https://doi.org/10.5281/zenodo.4285911>), as the performance is still an issue we would like to improve on.

4.2 Visualizations of `datsteps()`: Histogram vs. Density

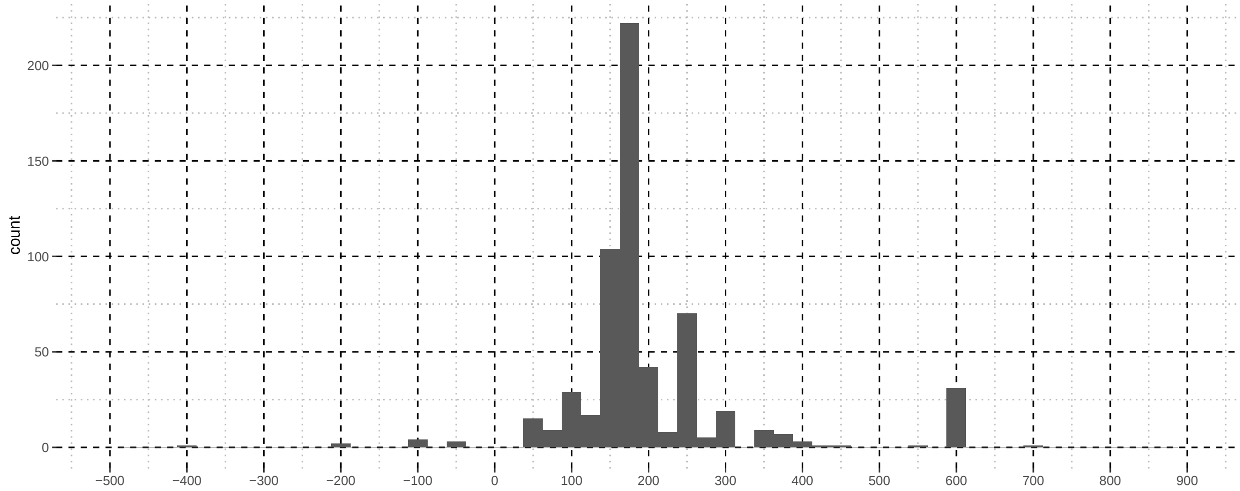
To determine the benefits of using other forms of visualization, we first want to introduce some problems arising from the use of histograms on this particular data set. Let us take a look at the distribution of the minimum and maximum values for each inscription, that we can easily display as a histogram:

```
inscr_Nicaea %>%
  select(ID, DAT_min, DAT_max) %>%
  reshape2::melt(id = "ID") %>%
  ggplot(aes(x = value, fill = variable)) + Plot_Theme +
  geom_histogram(binwidth = 25, position = "stack") +
  scale_fill_manual(name = "", values = bluegreen(2)) + Plot_Scale
```



Apparently, this distribution of upper and lower boundaries does not allow for getting a reasonable impression about the chronology of the inscriptions. The division into bins poses a well known problem, as the shape of the distribution may change depending on the starting point of the first bin and the `binwidth` parameter, i.e. the interval into which cases are grouped (Shennan 1997:27–28). A lot of inscriptions seem to be dated between large timespans, but we are not able to discern the overlaps. We can again try the same visualization with the mean dating of each object.

```
inscr_Nicaea %>%
  transmute(ID, DAT_mean = ((DAT_min+DAT_max)/2)) %>%
  ggplot(aes(x = DAT_mean)) + Plot_Theme +
  geom_histogram(binwidth = 25, position = "stack") + Plot_Scale
```



There is a peak in the second century AD, but we cannot be sure if this is true or due to the mean of the Roman Imperial Period (at 182 AD) falling into this century, as there are at least 217 inscriptions dated only roughly to this timespan. Let us now examine the visualization as a density graph with the output of `datsteps()`. In order to clarify which format needs to be processed to `datsteps()`, we select the table again. The `datsteps()`-function requires a `data.frame` with four variables in the following order: Identifier, variable, minimum dating, maximum dating. This can easily be achieved with the `select()`-function from the `tidyverse`.

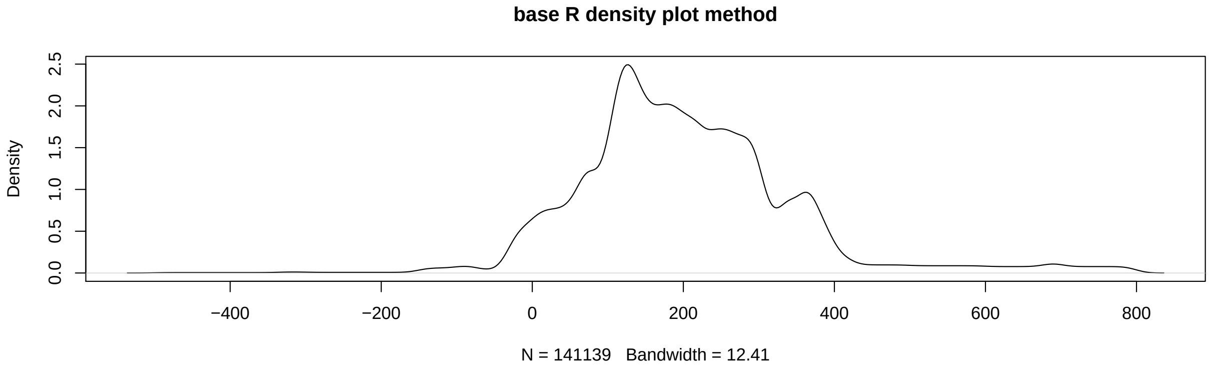
```
Nicaea_steps <- inscr_Nicaea %>%
  select(ID, Language, DAT_min, DAT_max) %>%
  datsteps(stepsize = "auto")
```

```
## [1] "Using stepsize = 1 (auto)."
```

The easiest way to visualize the data is the `density()` function together with the base R plot method:

```
plot(density(Nicaea_steps$DAT_step,
             weights = Nicaea_steps$weight),
     main = "base R density plot method")
```

```
## Warning in density.default(Nicaea_steps$DAT_step, weights =
## Nicaea_steps$weight): sum(weights) != 1 -- will not get true density
```

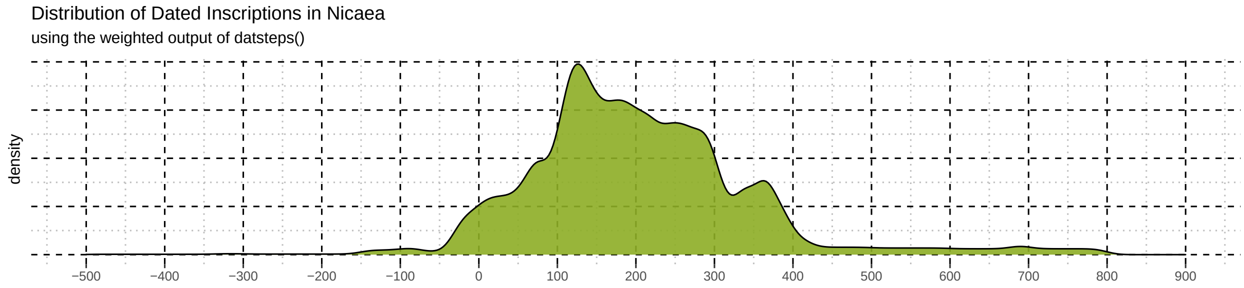


The warning returned by `density()` informs us that for a correct usage of kernel density estimation, the sum of weights needs to be 1. We integrated a function into the datplot-package, that can either scale the weights across all objects (`scaleweight(..., var = "all")`), or scale them in such a way that the sum of weights for each subgroup determined by the variable equals 1 (`scaleweight(..., var = 2)`), which is needed for separate visualization. In this case, we are not looking at the variable, and will therefore scale across all rows:

```
Nicaea_steps <- scaleweight(Nicaea_steps, var = "all")
```

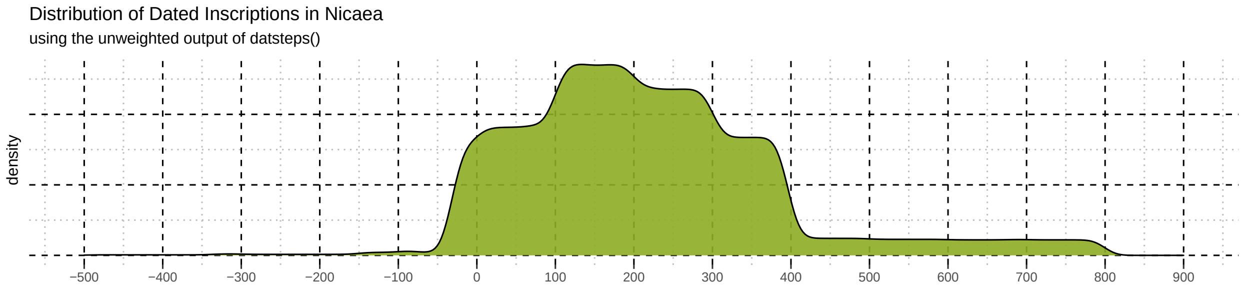
Visualizing the output with `ggplot2` holds more possibilities for styling. First, we use the bandwidth automatically determined by the plot method itself, as already done above as well:

```
ggplot(Nicaea_steps, aes(x = DAT_step, weight = weight)) + Plot_Theme +
  geom_density(alpha = 0.9, fill = bluegreen(1)) + No_Y +
  labs(x = "Dating", title = "Distribution of Dated Inscriptions in Nicaea",
       subtitle = "using the weighted output of datsteps()") + Plot_Scale
```



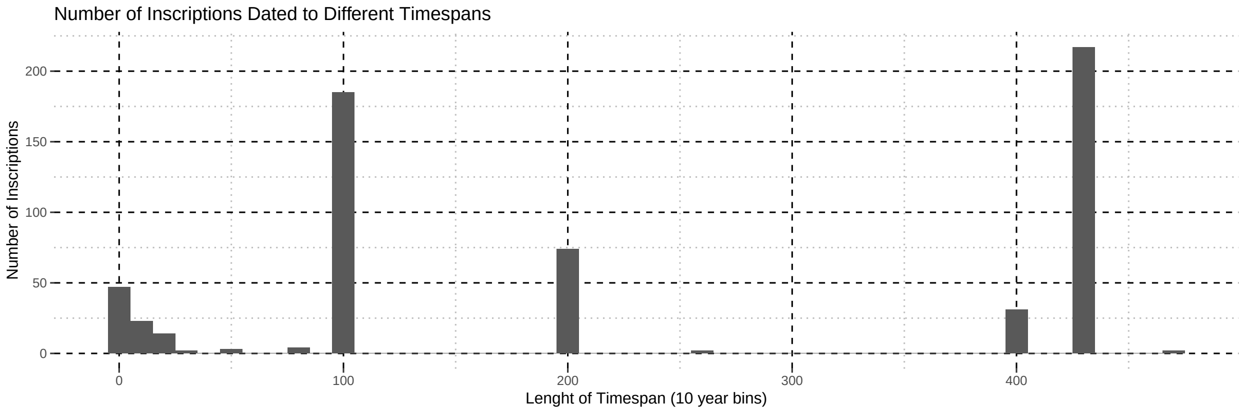
Since `ggplot2`s `geom_density()` uses the base R `density()`-function, the curve looks exactly the same. Plotting this same graph without including the *weight*-measure determined by **datplot** returns the following result:

```
ggplot(Nicaea_steps, aes(x = DAT_step)) + Plot_Theme +
  geom_density(alpha = 0.9, fill = bluegreen(1)) + No_Y +
  labs(x = "Dating", title = "Distribution of Dated Inscriptions in Nicaea",
       subtitle = "using the unweighted output of datsteps()") + Plot_Scale
```



We can observe a substantial difference in the two outputs. The first graph – with the *weight*-measure – shows a distinct peak in the beginning of the second century, as well as smaller peaks and valleys in the remaining curve. The second graph on the other hand – without the *weight*-measure – shows a broad distribution without obvious peaks, but instead several plateaus, the highest of which is spanning almost the entire second century. This is because of the inscriptions dated to a single year or a short span of time, which in the first graph contribute more to the resulting curve. We can quickly examine the distribution of timespans:

```
inscr_Nicaea %>%
  select(ID, Dating, Language, DAT_min, DAT_max) %>%
  mutate(timespan = DAT_max-DAT_min) %>%
  ggplot(aes(x = timespan)) + Plot_Theme +
  geom_histogram(binwidth = 10) +
  labs(title = "Number of Inscriptions Dated to Different Timespans",
       x = "Lenght of Timespan (10 year bins)", y = "Number of Inscriptions")
```

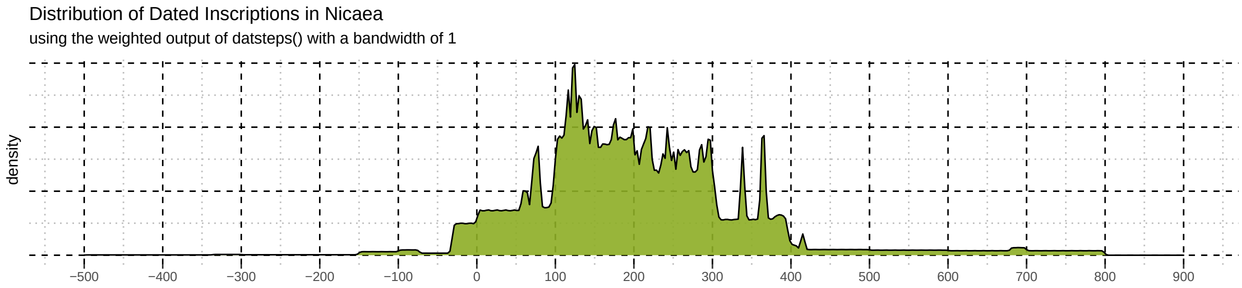


This helps us understand the placement of the more dominant peaks. In the unweighted graph, the large amount of inscriptions dated to wider ranges appears as background noise obscuring the possibly significant peaks in well dated inscriptions in the 1st century BC as well as the 2nd and 4th centuries AD, for example. **datplot** assigns closely dated objects with lower timespans a higher weight which subsequently leads to their greater importance for the density graphs. This might not always turn out useful in archaeological contexts, depending on the objects examined and the heterogeneity of the data. As stated above, it might be therefore often useful to look at both, the weighted and the unweighted outputs, as they inform the viewer of different occurrences. In the case of inscriptions, a large amount of objects dated to small time spans might point to certain important occurrences of the region's history.

4.3 Choosing Bandwidth

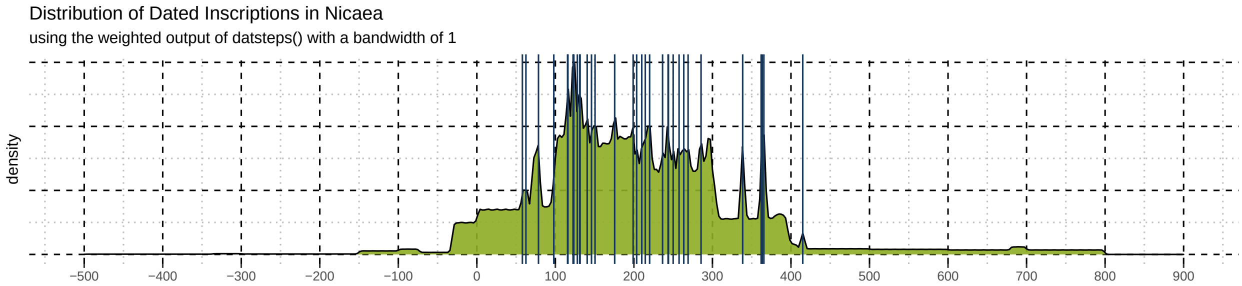
Furthermore, it is possible to assign the bandwidth of the density graph. The bandwidth controls how far the curve will be 'smoothed' by the plot. If we select the bandwidth according to our stepsize, the output changes as follows:

```
ggplot(Nicaea_steps, aes(x = DAT_step, weight = weight)) + Plot_Theme +
  geom_density(alpha = 0.9, fill = bluegreen(1), bw = 1) + Plot_Scale + No_Y +
  labs(x = "Dating", title = "Distribution of Dated Inscriptions in Nicaea",
       subtitle = "using the weighted output of datsteps() with a bandwidth of 1")
```

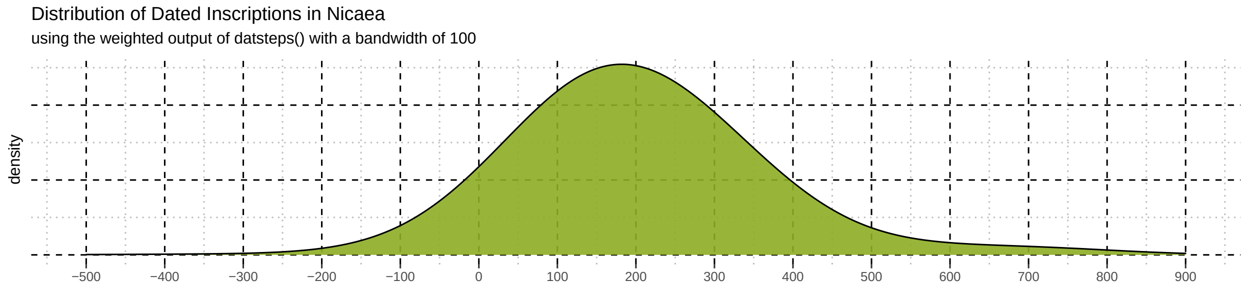


The inscriptions with the most precise dating, i.e. less than or equal to 5 years, were dated around the years 244, 98, 78, 62, 58, 415, 366, 364, 362, 338, 286, 269, 264, 258, 250, 236, 220, 214, 210, 204, 199, 176, 150, 146, 140, 132, 131, 128, 124, 123, 122, 116 (all AD), and cause the most prominent peaks in this distribution. Marking them in the graph, while producing a somewhat confusing visualization, may serve to illustrate their influence and is certainly a good way to examine the data along the way in order to better understand the distribution:

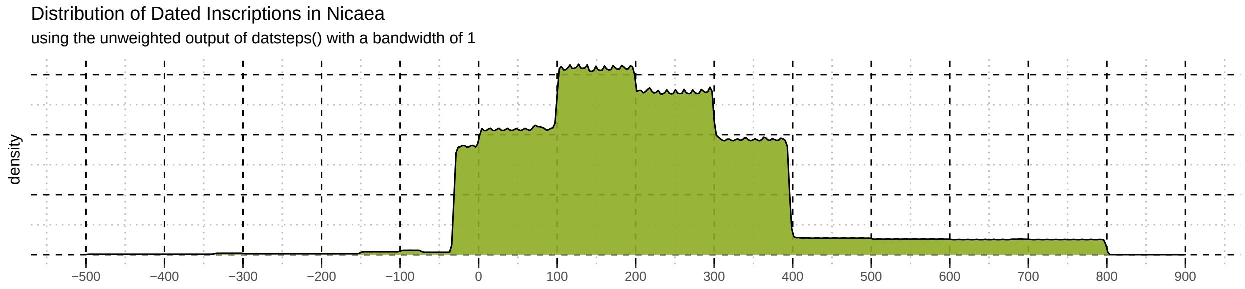
```
precise_inscr <- which(c(inscr_Nicaea$DAT_max - inscr_Nicaea$DAT_min) < 6)
precise_inscr <- (inscr_Nicaea$DAT_min[precise_inscr] +
                    inscr_Nicaea$DAT_max[precise_inscr]) / 2
ggplot(Nicaea_steps, aes(x = DAT_step, weight = weight)) + Plot_Theme +
  geom_density(alpha = 0.9, fill = bluegreen(1), bw = 1) + Plot_Scale + No_Y +
  labs(x = "Dating", title = "Distribution of Dated Inscriptions in Nicaea",
       subtitle = "using the weighted output of datsteps() with a bandwidth of 1") +
  geom_vline(xintercept = precise_inscr, color = bluegreen(2)[2])
```

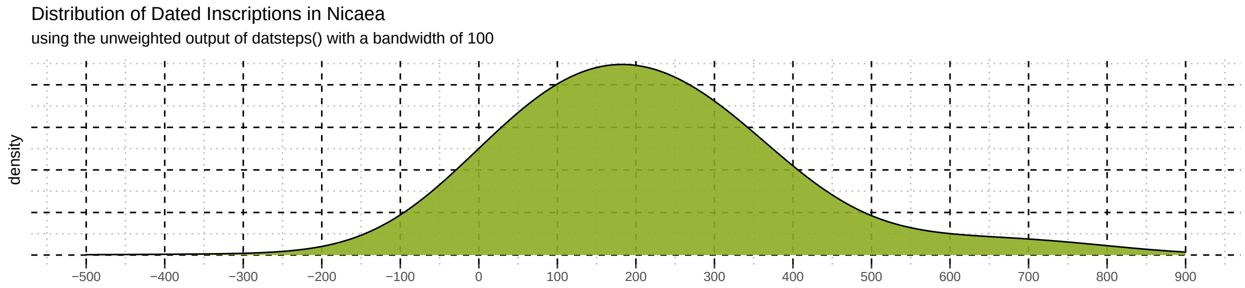


Likewise, selecting a larger bandwidth will yield a smoother curve, obscuring potential peaks that may otherwise point to historical events or increased epigraphic activity. The only thing changed in the code compared to the graphs above is the bandwidth parameter:



The unweighted graphs with the same bandwidths (1 and 100 respectively) present themselves as follows:





As we can see, a larger bandwidth obscures the information we can gather from the distribution. In practical terms, the density estimate further distributes each kernel over an amount of years equal to the bandwidth-parameter (see ‘Kernel Density Estimation and its Previous Uses in Archaeology’ in the paper). This means that an inscription dated at AD 50 would be most strongly influencing the curve at that particular year, but also (with a uniform distribution, and therefore increasingly less so) the years surrounding it. Employing it in some way seems reasonable to us, as in archaeological terms, we are actually very rarely able to speak about developments during single years, but tend to think in more broad terms such as quarter centuries – depending on the discipline and matter of study, obviously. Furthermore, if we view any object as an ‘event’ happening, there are always steps leading to and from this event. Be it the occasions referenced in the inscription, a pottery style becoming widely adopted, or the decision to mint a new coin. Likewise, the life of any artifact does never end with its production date. It is further seen, used and circulated. Therefore, each object can be considered to have some influence on or be influenced by the years before and after its set date. Greatly depending on the matter of study, the bandwidth can reflect this idea. At the same time, there is another issue of uncertainty that the bandwidth parameter addresses well. This may hold true especially for stylistically dated objects, such as pottery. If one determines the production of a vessel to have occurred between 425 and 400 BC, in most cases this is already a probability-statement. Rarely can we have the kind of certainty required for respecting the bounds set by absolute dates. Is there really any guarantee that it could not have been produced in 426, 427 or 399 BC? The bandwidth does spread this probability a bit further, and researchers can determine how much further it should be spread by adjusting the parameter accordingly. Having said that, we recommend making an informed decision about the use of bandwidth. In our opinion, the plot should be aesthetically pleasing without giving up information. This greatly depends on the data used, and therefore specific values cannot be universally recommended.

Setting the stepsize to a quarter century achieves an equilibrium between exactly and more loosely dated inscriptions, and seems reasonable for spotting longterm trends in the case of the Bithynia-data set.

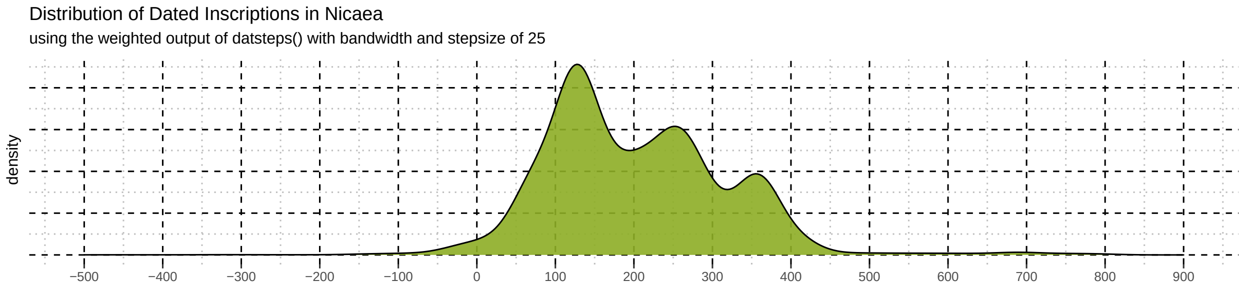
```
Nicaea_steps <- inscr_Nicaea %>%
  select(ID, Language, DAT_min, DAT_max) %>%
  datsteps(stepsize = 25)
```

```
## Warning in get.weights(DAT_mat[, "datmin"], DAT_mat[, "datmax"]): Warning:
## DAT_min and DAT_max at Index: 284, 525, 526, 530, 536, 579, 584, 587, 588, 590,
## 591, 592, 599) have the same value! Is this correct? Please check the table for
## possible errors.
```

```
## Warning in create.sub.objects(DAT_mat, stepsize): stepsize is larger than the
## range of the closest dated object at Index = 223, 226, 227, 228, 229, 261,
## 262, 263, 264, 265, 266, 267, 269, 271, 272, 273, 274, 275, 276, 277, 278,
## 279, 281, 284, 290, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378,
## 379, 380, 517, 518, 519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 530,
## 531, 532, 533, 534, 535, 536, 537, 579, 580, 582, 583, 584, 585, 586, 587, 588,
## 589, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 600, 601, 602, 603). For
## information see documentation of get.step.sequence().
```

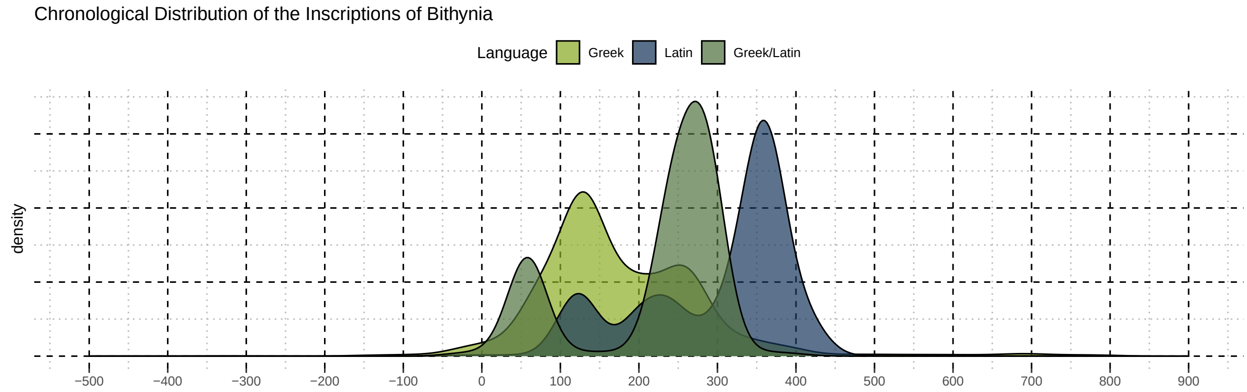
Now, datplot also informs us of the discrepancy between closest dated objects and stepsize and lists the affected indexes (row numbers) from the original data. For the plot, the value used as a stepsize can be passed on to the bandwidths argument in `geom_density()` using the corresponding attribute of the `data.frame` returned by `datsteps()`:

```
ggplot(scaleweight(Nicaea_steps, var = "all"),
       aes(x = DAT_step, weight = weight)) + Plot_Theme +
  geom_density(alpha = 0.9, fill = bluegreen(1),
               bw = attributes(Nicaea_steps)$stepsize) + No_Y +
  labs(x = "Dating", title = "Distribution of Dated Inscriptions in Nicaea",
       subtitle = paste("using the weighted output of datsteps() ",
                        "with bandwidth and stepsize of ",
                        attributes(Nicaea_steps)$stepsize,
                        sep = "")) + Plot_Scale
```



As mentioned before, we can also produce a plot separating the variables. For this, we scale the weights along the variable:

```
ggplot(scaleweight(Nicaea_steps, var = 2),
       aes(x = DAT_step, fill = variable, weight = weight)) + Plot_Theme +
  geom_density(alpha = 0.7, bw = 25) +  Plot_Scale + No_Y +
  labs(title = "Chronological Distribution of the Inscriptions of Bithynia") +
  scale_fill_manual(values = bluegreen(3)[c(1, 3, 2)], name = "Language")
```



One needs to be cautious when interpreting the output in this case, as has already been argued in the paper itself. The distribution does not inform us about the quantity of the inscriptions, but about their ‘datability’. We can learn that Latin inscriptions are generally dated later than the Greek inscriptions, and that the highest overlap of Greek and Latin inscriptions is set in the mid 3rd century. A look at the actual quantitative relationship of the languages helps to understand the sometimes deceptive nature of this visualization. Generally speaking, this kind of analysis along variables works better with groups of more equal sizes.

```
table(inscr_Nicaea$Language)
```

```
##
##       Greek       Latin Greek/Latin
##         554          38          12
```

This gives us a more direct impression of how the density graph works, though the output when setting the bandwidth to the same value as stepsize is essentially a histogram with lines instead of bars. Many approaches to aoristic analysis indeed prefer line graphs or histograms (Ratcliffe 2000:672–673 and Fig. 2–3; Mischka 2004).

4.4 Combining Density Graph and Histogram

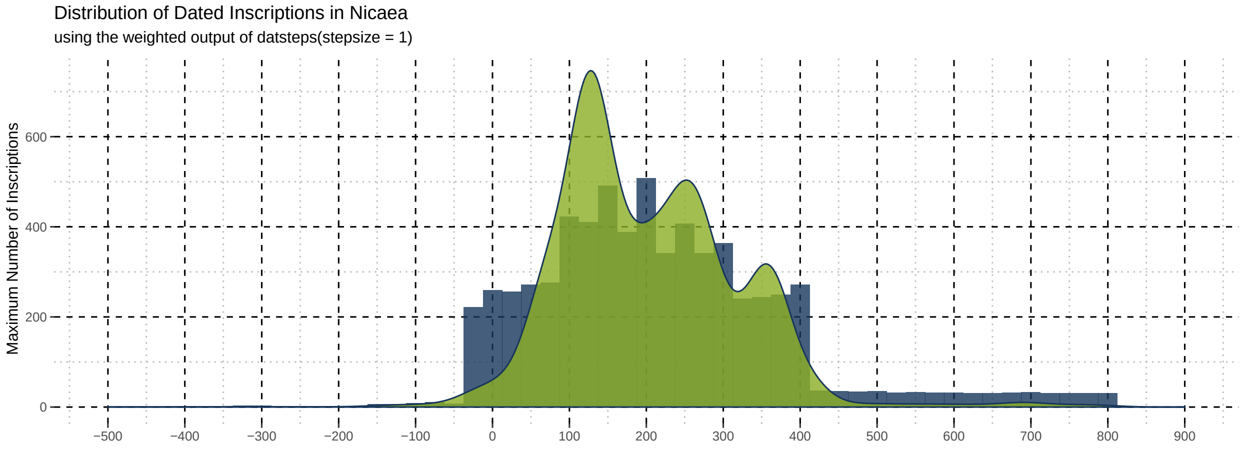
In many cases, archaeologists may want to convey also the number of finds dated to each year. This is not possible, as we do not have this kind of information. However, the maximum number of finds for each year may well be plotted alongside, combining the density plots with a histogram. The histogram – when employing the dating steps produced by `datsteps()` – will display the maximum possible count of objects for each year. There is no way to discern the actual number of objects from the graph.

We included a way of determining the value along which the density graph needs to be scaled in order to display it in the same graph as the corresponding histogram:

```
binwidth <- attributes(Nicaea_steps)$stepsize
scalevalue <- get.histogramscale(Nicaea_steps, binwidth = binwidth)
```

With this value, the density curve can be scaled in order to be visible in the histogram-plot. Keep in mind that the y-axis is not meaningful for the kernel density estimate.

```
ggplot(Nicaea_steps, aes(x = DAT_step)) + Plot_Theme + Plot_Scale +
  geom_histogram(alpha = 0.8, binwidth = binwidth, fill = bluegreen(2)[2]) +
  stat_density(alpha = 0.8,
               color = bluegreen(2)[2],
               fill = bluegreen(2)[1],
               bw = binwidth,
               aes(y = (..density.. * scalevalue), weight = weight)) +
  labs(x = "Dating", y = "Maximum Number of Inscriptions",
       title = "Distribution of Dated Inscriptions in Nicaea",
       subtitle = "using the weighted output of datsteps(stepsize = 1)",
       caption = paste("(redacted), ",
                       attributes(inscr_clean)$source, sep = ""))
```



(redacted), Data: https://inscriptions.packhum.org/ and
B. Weissová, Regional Economy, Settlement Patterns and the Road System in Bithynia
(4th century BC – 6th century AD) (Diss. FU Berlin 2019)

In total, a combination of weighted and unweighted density graphs for the Inscriptions of Nicaea yields the most reasonable results, informing us of the distribution of closely dated inscriptions as well as loosely dated ones. We generally see a uniform distribution in the *quantity* of inscriptions between the Hellenistic and late Antique periods, but can also assert several points in time that have a greater overlap of inscriptions, possibly pointing to historical events that are responsible for such peaks in "datability" and the accumulation of well dated inscriptions in around certain years.

5 Comparison with Visualizations in the Previous Study

Importing the one-hot-table[1], that has been used in the previous study (Weissova 2019), we can visualize an easy comparison between the two methods.

```
# The table with the roman imperial period starting 31 BC
onehot <- readRDS("data/prep/weissova_2019_onehot_correct.rds")

# The table with the roman imperial period starting AD 1
onehot_imp_AD <- readRDS("data/prep/weissova_2019_onehot_impAD.rds")
```

```
inscr_Nicaea %>%
  select(ID, DAT_min, DAT_max) %>% na.omit() %>%
  left_join(onehot[, c(3:12, 17)], by = "ID") %>%
  sample_n(5)
```

[1] The setup of this "one-hot-table" is explained in the paper. One-hot-encoding refers to the translation of nominal variables (in this case 'centuries') into a binary format (1 for presence, 0 for absence of said variable), which is in many cases more easily machine readable.

ID	DAT_min	DAT_max	5BC	4BC	3BC	2BC	1BC	1AD	2AD	3AD	4AD	5AD
I_1393	100	299	0	0	0	0	0	0	1	1	0	0
I_1176	200	299	0	0	0	0	0	0	0	1	0	0
I_895	-31	395	0	0	0	0	1	1	1	1	1	0
I_1457	1	199	0	0	0	0	0	1	1	0	0	0
I_955	-31	395	0	0	0	0	1	1	1	1	1	0

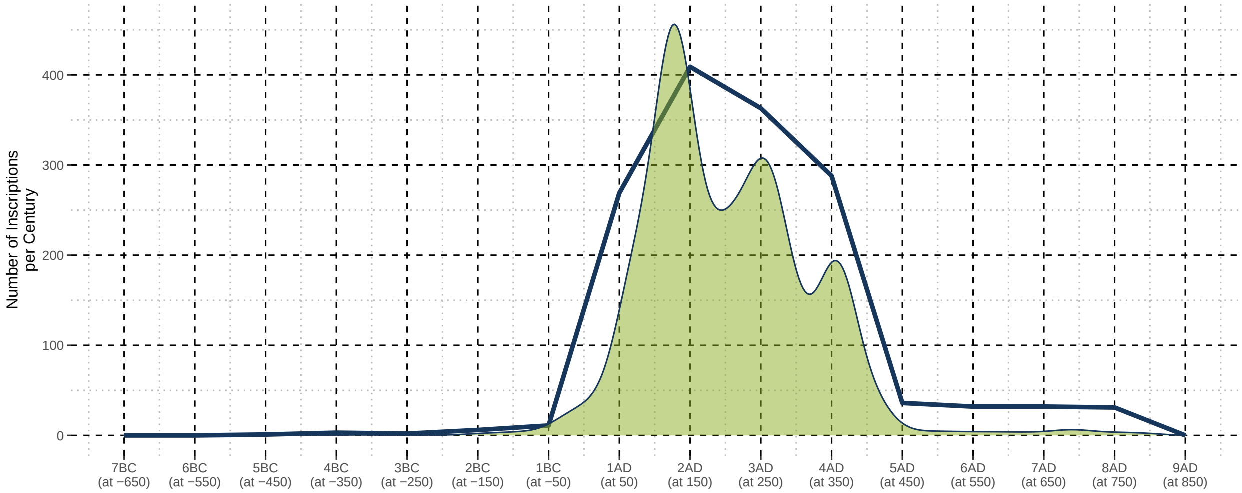
In order to be able to place both data sets on a shared x-axis, we need to supply numeric values for the centuries. We chose to display each century's axis-tick in its middle:

```
replacetable <- t(rbind(colnames(onehot)[1:16], seq(-650, 850, 100)))
replacetable <- as.data.frame(replacetable)
colnames(replacetable) <- c("variable", "value")
replacetable$value <- as.numeric(replacetable$value)
```

variable	7BC	6BC	5BC	4BC	3BC	2BC	1BC	1AD	2AD	3AD	4AD	5AD	6AD	7AD	8AD	9AD
value	-650	-550	-450	-350	-250	-150	-50	50	150	250	350	450	550	650	750	850

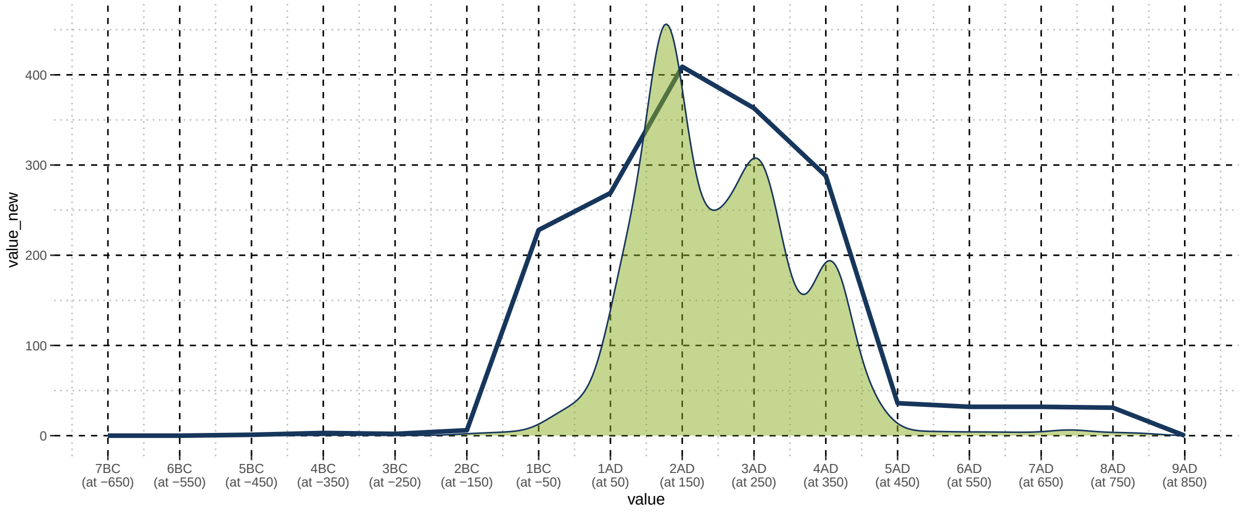
With this concordance, we can substitute the values in our one-hot-table and display it together with the scaled density. The graph shows the chronological distribution of inscriptions in Nicaea on a time scale divided into centuries. In this case, the beginning of Roman Imperial Period was counted as the 1st c. AD for the blue line, while the density curve uses the beginning of the Roman Imperial Period at 31 BC with a stepsize and bandwidth of 25.

```
onehot_imp_AD %>%
  left_join(inscr_clean[, c(3, 1)], by = "ID") %>%
  filter(Location == "Nicaea") %>%
  reshape2::melt()  %>%
  group_by(Location, variable) %>%
  summarise(value_new = sum(value)) %>%
  left_join(replacetable, by = "variable") %>%
  ggplot() + Plot_Theme +
  geom_line(aes(x = value, y = value_new),
            color = bluegreen(2)[2], lwd = 1.5) +
  scale_fill_manual(guide = FALSE) +
  scale_x_continuous(breaks = seq(-650, 850, 100),
                     labels = paste(unique(replacetable$variable), "\n(at ",
                                    seq(-650, 850, 100), ")", sep = "")) +
  geom_density(data  = Nicaea_steps, bw = attributes(Nicaea_steps)$stepsize,
               aes(x = DAT_step, y = ..density.. * 100000, weight = weight),
               fill = bluegreen(2)[1], color = bluegreen(2)[2], alpha = 0.5) +
  labs(x = "", y = "Number of Inscriptions\nper Century")
```



Note that we had to choose a scale value that seemed fine in the visualization. The values on the y-axis on the density curve are not meaningful for further analysis of the output, and therefore it is not problematic to scale the curve accordingly.

```
onehot %>%
  left_join(inscr_clean[, c(3, 1)], by = "ID") %>%
  filter(Location == "Nicaea") %>%
  reshape2::melt() %>%
  group_by(Location, variable) %>%
  summarise(value_new = sum(value)) %>%
  left_join(replacetable, by = "variable") %>%
  ggplot() + Plot_Theme +
  geom_line(aes(x = value, y = value_new), color = bluegreen(2)[2], lwd = 1.5) +
  scale_fill_manual(guide = FALSE) +
  scale_x_continuous(breaks = seq(-650, 850, 100),
                     labels = paste(unique(replacetable$variable),
                                    "\n(at ", seq(-650, 850, 100), ")", sep = "")) +
  geom_density(data = Nicaea_steps, bw = attributes(Nicaea_steps)$stepsize,
               aes(x = DAT_step, y = ..density..*100000, weight = weight),
               fill = bluegreen(2)[1], color = bluegreen(2)[2], alpha = 0.5)
```



Compared to the approach taken in the original publication (Weissova 2019), a chronological kernel density estimate produced with the output of datplot seems to be more informative about the actual distribution of inscriptions in Bithynia across time. Each density graph that we have shown in this supplement highlights several aspects that can be teased out of the very same data that was used for the original line graphs, which only inform us of an almost uniform distribution which peaks in the second century AD. **datplot** can be very useful in further exploring a data set as well as visualizing it in terms of accumulations of objects in time, that may point to occurrences, events and developments.

6 References

Mischka, Doris
2004 Aoristische Analyse in der Archäologie. Archäologische Informationen: Mitteilungen zur Ur- und Frühgeschichte 27(2):233–243.

Shennan, Stephen
1997 Quantifying Archaeology. Edinburgh University Press, Edinburgh.

Ratcliffe, Jerry H.
2000 Aoristic Analysis: the Spatial Interpretation of Unspecific Temporal Events. International Journal of Geographical Information Science 14(7):669–679.

Weissova, Barbora
2019 Regional Economy, Settlement Patterns and the Road System in Bithynia (4th century BC - 6th century AD). Volume 1 Text. Electronic document, https://refubium.fu-berlin.de/bitstream/handle/fub188/23730/Thesis_Weissova_Volume_1.pdf?sequence=4&isAllowed=y, accessed July 19, 2020.

Weissova, Barbora, Robin Brigand, and Silvia Polla
2019 "The Hinterland of Nikaia/Nicaea/Iznik: Analyzing the Hellenistic Roman and Late Antique Bithynian Landscape Through Remote Sensing and GIS Techniques." eTopoi Journal for Ancient Studies 8:21–49. https://doi.org/10.17169/REFUBIUM-25618.