

lydemapr: an R package to map *Lycorma delicatula*

Sebastiano De Bona*

Matthew R. Helmus†

19 May 2025

Introduction

The Spotted lanternfly (*Lycorma delicatula*, White 1841) is an agricultural pest native of China and Southeast Asia, first discovered in the United states in 2014 in Berks County, PA. Since then, this planthopper has spread throughout the Mid-Atlantic and Midwest regions of the country, threatening the wine and fruit industry and damaging ornamental trees.

Since its first discovery, many sources have collected data on the presence/absence and population density of this species in order to monitor its spread and impact. The **lydemapr** package contains two anonymized datasets (at 1 km² and 10 km² resolution) resulting from an effort to combine, organize, and aggregate all available sources of data. In addition, this package contains useful functions to visualize the data within R.

The **lydemapr** package was built with the intent to increase accessibility to key data on this species of interest, and to improve reproducibility and consistency of modeling efforts.

We are constantly looking to expand the data sources to have a full representation of SLF's presence and abundance in the US. If you wish to contribute to this effort please contact the package authors.

```
# attaching necessary packages
library(lydemapr)
library(sf)
library(tidyverse)
library(tigris)
```

Data Summary

To begin, let's take a look at the data structure:

```
head(lydemapr::lyde)
```

```
## # A tibble: 6 x 14
##   source year bio_year latitude longitude state lyde_present lyde_established
##   <chr>   <dbl>   <dbl>   <dbl>      <dbl> <chr>   <lgl>      <lgl>
## 1 inat   2015     2015    40.4     -75.7 PA     TRUE      FALSE
## 2 inat   2016     2016    40.3     -75.6 PA     TRUE      FALSE
## 3 inat   2016     2016    40.4     -75.5 PA     TRUE      FALSE
## 4 inat   2016     2016    40.4     -75.6 PA     TRUE      FALSE
## 5 inat   2016     2016    40.4     -75.7 PA     TRUE      FALSE
```

*Temple University, sebastiano.debona@gmail.com

†Temple University, mrhelmus@temple.edu

```
## 6 inat      2016      2016      40.5      -75.6 PA      TRUE      FALSE
## # i 6 more variables: lyde_density <fct>, source_agency <chr>,
## #   collection_method <chr>, pointID <chr>, rounded_longitude_10k <dbl>,
## #   rounded_latitude_10k <dbl>
```

Each data point contains information on its source and specific dataset of origin (“source_agency”). The data is organized by year (specified as both calendar “year” and “bio_year”, running from May 1st to April 30th), coordinates, and state. Additional columns define whether SLF was found during the survey in that location (even as an anecdotal individual record, “lyde_present”), whether an established population was found there (“lyde_established”), and what the estimated population density of SLF was there (“lyde_density”). For additional information on the variables included, please consult the help file associated with the data by typing `?lyde` in the RStudio console. A Metadata file can also be found in the compressed folder `lyde_data.zip` contained in `download_data/`.

The package function `lyde_summary()` breaks the data down into a quick summary, with data organized by different axes. We can take a look at the data split across year and States. It’s important to notice that the data is arranged yearly according to the **biological year** of SLF, and not calendar year. This allows for the appropriate inclusion of egg masses discovered during the winter months which were laid during the previous calendar year’s summer/fall.

```
# data by Year and State
knitr::kable(lyde_summary(year_type = "biological"))
```

```
## Warning: 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
```

```
## Warning: 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
```

	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024
AZ	0	0	0	0	0	10	139	120	205	672	520
CA	0	0	0	0	0	0	0	0	1	1	0
CT	0	0	0	0	0	4	2094	1442	1660	1894	351
DC	0	0	0	0	8	21	10	5	0	47	0
DE	0	0	0	0	1075	2208	4546	6962	6043	6427	3083
FL	0	0	0	0	0	0	0	1	0	0	0
IL	0	0	0	0	0	0	0	0	0	16	0
IN	0	0	0	0	79	101	103	508	195	282	5
KS	0	0	0	0	0	0	0	21	0	0	0
KY	0	0	0	0	0	3	2	20	165	168	16
MA	0	0	0	0	0	0	893	2859	2097	3949	2995
MD	0	0	0	1	39	2399	17408	4734	1656	3231	1029
ME	0	0	0	0	0	0	0	20	85	37	28
MI	0	0	0	0	0	0	1	133	307	699	314
MO	0	0	0	0	0	15	18	0	0	84	0
NC	0	0	0	0	0	14067	5	110	4858	2024	453
NH	0	0	0	0	0	0	0	0	60	22	3
NJ	0	0	0	0	2443	9529	13075	83705	39706	2411	488
NM	0	0	0	0	0	0	10	28	26	21	17
NY	0	0	0	0	18474	27046	18228	12826	22526	19495	3557

	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024
OH	0	0	0	0	0	0	681	575	1227	1464	165
OR	0	0	0	0	0	0	92	15	73	3	337
PA	372	7677	9269	9231	77055	150180	90476	69436	82517	98580	30620
RI	0	0	0	0	0	0	45	18	285	777	1453
SC	0	0	0	0	0	2	7	49	78	80	38
TN	0	0	0	0	0	0	0	1	0	1166	1138
TX	0	0	0	0	0	0	0	0	150	472	0
UT	0	0	0	0	0	0	1	0	0	0	0
VA	0	0	0	2	1523	4353	4100	2574	3977	6235	1150
VT	0	0	0	0	0	0	0	2	25	1	0
WI	0	0	0	0	0	0	0	0	0	1	0
WV	0	0	0	0	3	995	2368	2101	1905	1621	702

Maps of the Spread of SLF

Two functions allow the user to plot the data: `map_spread()` and `map_yearly`.

The first function produces a snapshot of the SLF spread in the United States, with reference to the sampling effort associated with surveying the spread. Surveys finding an established population are plotted on the map as filled tiles, color coded by the year of first discovery. Surveys finding no established population are plotted as grey tiles.

As the plotting of the data might take a long time to display within R, we encourage the user to assign the map and save it as a pdf instead, like we show below.

```
# assigning the map
map_1 <- map_spread()
```

The map can be saved as a pdf file at high resolution.

```
# saving the map as a pdf
pdf("Map_spread.pdf", width = 7.5, height = 8)
map_1
dev.off()
```

```
# If executing this line while running the vignette manually,
# be advised that it might take a considerable amount of time
# for the map to be displayed.
# It's advised to visualize the pdf file saved above.
map_1
```

The default function displays data aggregated at the 10km² (Figure 1). The function can be customized to show the data at higher spatial resolution (1k²), by setting the function option `resolution` to "1k". This will take considerably longer, so saving the result as a pdf is preferable in this instance as well.

```
map_2 <- map_spread(resolution = "1k")
```

```
pdf("Map_spread_1k.pdf", width = 7.5, height = 8)
map_2
dev.off()
```

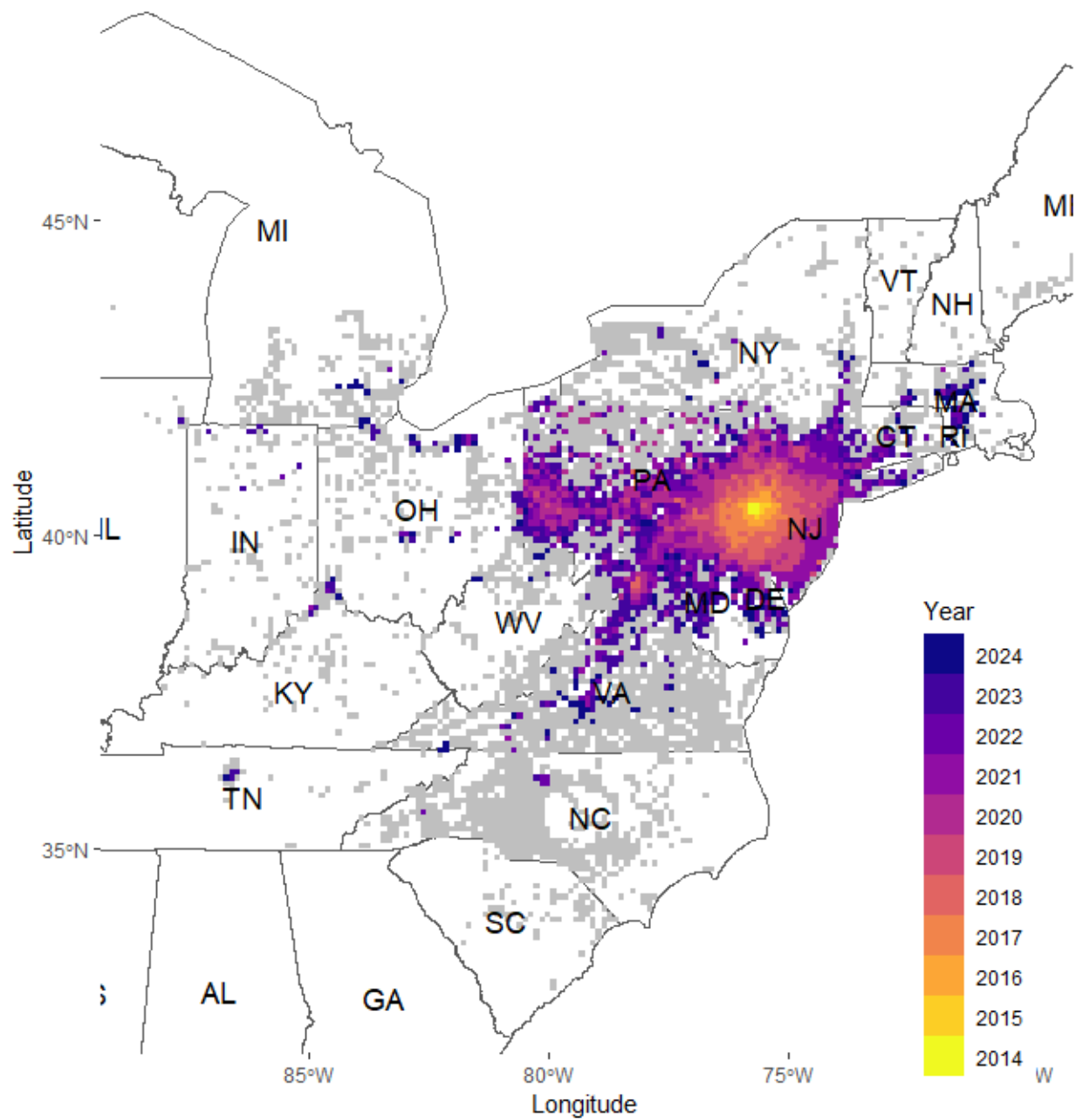


Figure 1: Output of the `map_spread()` function, plotted at the 10km resolution

```
# If executing this line while running the vignette manually,
# be advised that it might take a considerable amount of time
# for the map to be displayed.
# It's advised to visualize the pdf file saved above.
map_2
```

The function displays data in a slightly different fashion at the 1km² resolution (Figure 2). At 10km² the data is plotted at filled tiles. This improves the visualization by representing the grid in which the data is organized more clearly. As tiles of size 1km are much smaller, we prefer to display survey points at this resolution as points on the map.

If the user wishes to visualize the data for a smaller area of the United States, the function allows them to specify which area should be mapped, by setting the `zoom` variable to “custom” and specifying the boundaries of the mapped area through `xlim_coord` (longitude) and `ylim_coord` (latitude), as Longitude and Latitude coordinates using the WG84 projection. Here’s an example of how this can be achieved.

```
# assigning object
map_3 <- map_spread(resolution = "1k",
  zoom = "custom",
  xlim_coord = c(-78, -74),
  ylim_coord = c(38, 42))
```

```
# saving to pdf
pdf("Map_spread_1k_zoomed.pdf", width = 7.5, height = 8)
map_3
dev.off()
```

```
# If executing this line while running the vignette manually,
# be advised that it might take a considerable amount of time
# for the map to be displayed.
# It's advised to visualize the pdf file saved above.
map_3
```

The second function, `map_yearly()` allows the user to visualize the progression of SLF establishment, with a focus on the estimated population density through time. Note that the data here is not cumulative, meaning only data from a given year is shown in any given panel of the figure.

```
# running year-specific map
# assigning object
map_4 <- map_yearly(ncol = 3)
```

```
# saving to pdf
pdf("Map_yearly.pdf", width = 8, height = 9)
map_4
dev.off()
```

```
# If executing this line while running the vignette manually,
# be advised that it might take a considerable amount of time
# for the map to be displayed.
# It's advised to visualize the pdf file saved above.
# map_4
```

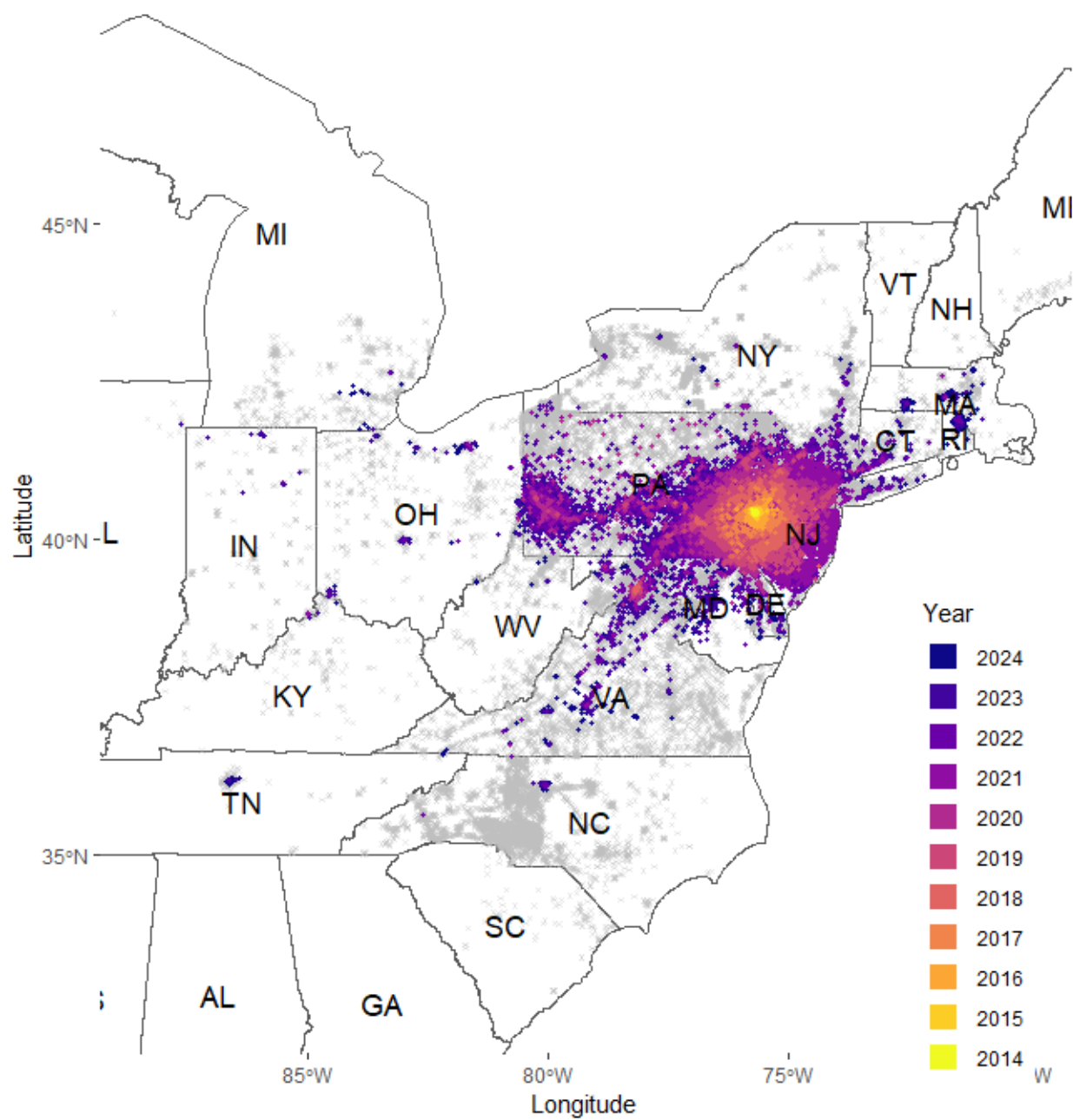


Figure 2: Output of the `map_spread()` function now plotted at a finer 1km resolution

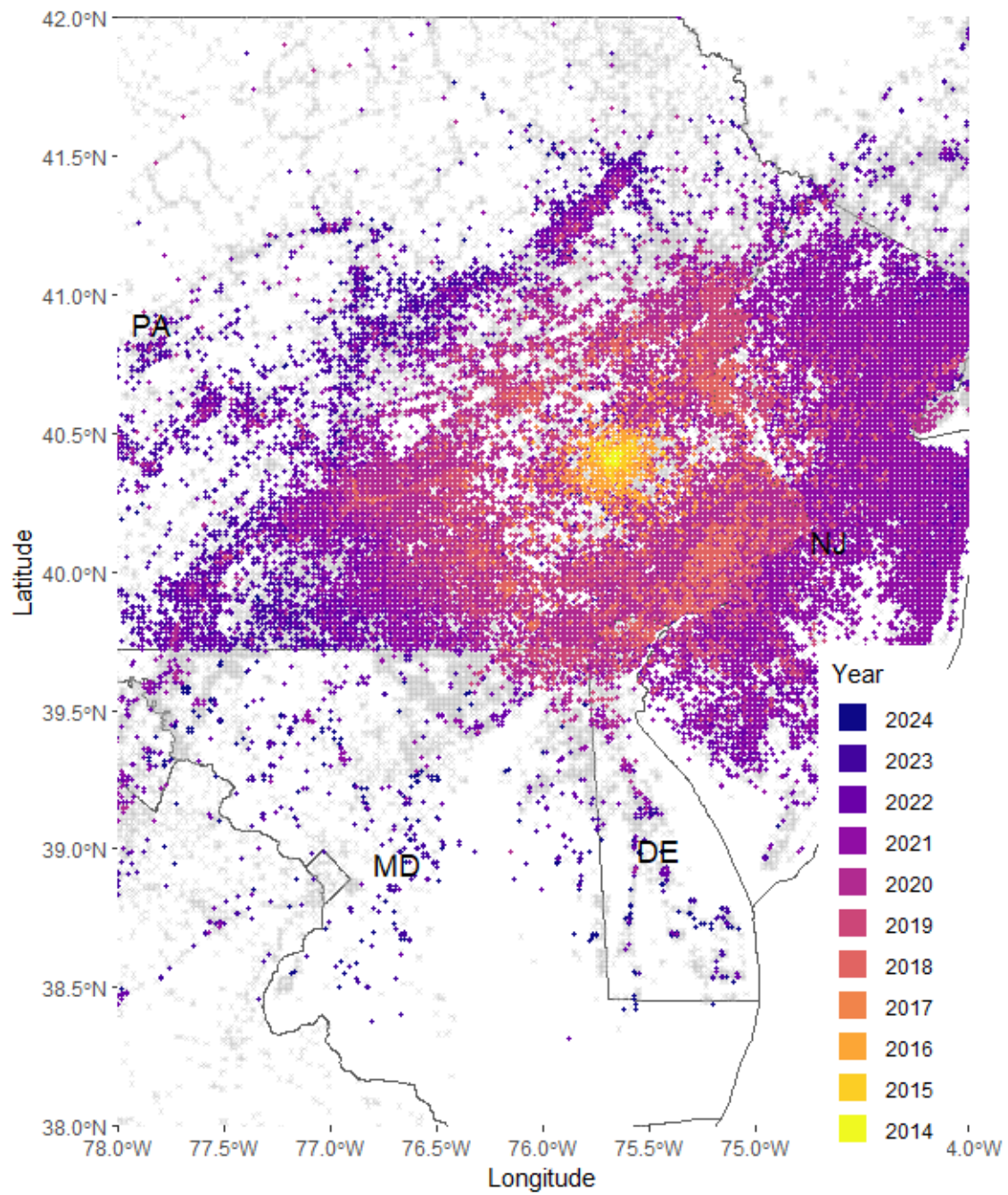


Figure 3: Zoomed area, focusing on the core of the invasion range