Exercise 2: Time series and annotation

Éric Marty & John Drake

May 23, 2022

A common visualization task in epidemiology and disease ecology is to plot a record of case counts over time, often referred to as the *epidemic curve*. The data are typically encoded with two variables, a date and the case count. An epidemic curve is just an ordinary line graph in two dimensions. However, more complicated versions of the epidemic curve are often useful, for instance showing two curves simultaneously. If those curves are of different magnitudes then they require different y-axes. Additionally, we often find it useful to annotate time series. This exercise teaches some of these tasks.

In 2003-2004 there was a large outbreak of measles in Niamey, Niger. Data on the number of cases in each of three communes aggregated to two week intervals is provided in the file `niamey.csv`.

```
data <- read.csv('../data/niamey.csv')
```

We can make a simple line plot of cases in Commune 1 using base R plotting functions as follows.

```
plot(data$Time, data$Commune1, type='l', xlab='Time')
```
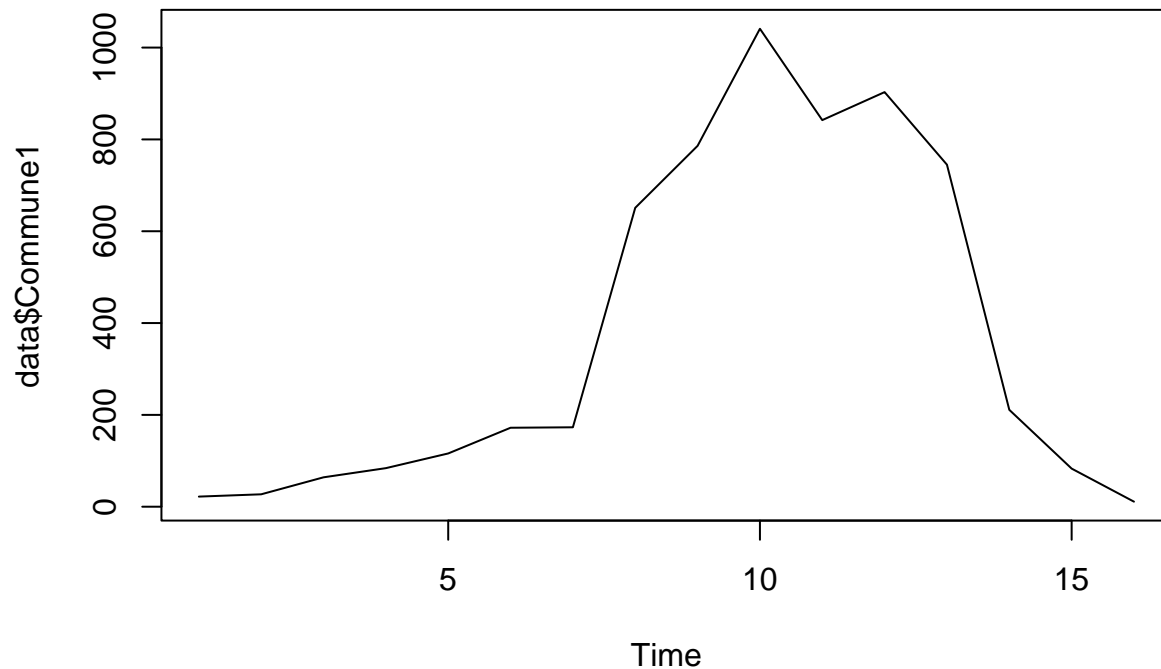

Exercise 1. Change the y-axis label, add a chart title, and place tick marks at all time points.

Exercise 2. Write a script or a function to find the time at which cases are at their highest. Use the R function `text` to annotate the figure with the word 'Peak' and a horizontal arrow.

Note: The R function `expression` is very useful for adding mathematical annotations to figures. A useful tutorial is available here.

Now we want to plot the data from all three communes. That is easy enough using the function `lines`.

```
plot(data$Time, data$Commune1, type='l', xlab='Time')
lines(data$Time, data$Commune2)
lines(data$Time, data$Commune3)
```
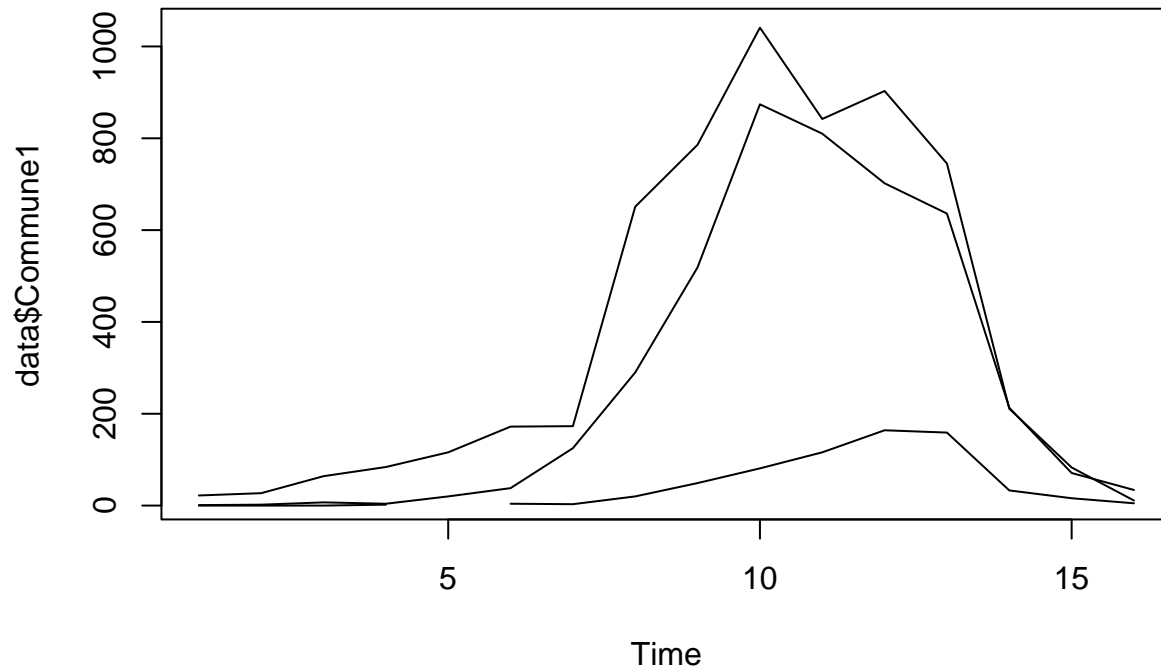

Clearly, there were fewer cases in Commune 3. If we want to compare the pattern in Commune 3 with those of Communes 1 and 2 it should be plotted on different axes. There are two main ways to do this. Both involve creating a "new" plot. The first way is to create a secondary y-axis. This is basically a new plot that is superimposed on the first plot. The second approach is to create two plots that are stacked with their x-axes aligned. Both approaches can be accomplished using the `par` function in R. Particularly, you will need to use the argument `new=TRUE`. To create the stacked plots, you will also need to use the argument `fig` to set up the figure region for each of your plots.

Exercise 3. Create a version of this figure with a secondary y-axis for Commune 3.

```
par(mar=c(5,4,4,4)+0.1)
plot(data$Time, data$Commune1, type='l', xlab='Time', ylab='Communes 1 and 2')
lines(data$Time, data$Commune2)
par(new=TRUE)
plot(data$Time, data$Commune3, type='l', col='red', xlab='', ylab='', axes=FALSE)
axis(4, col='red')
mtext('Commune 3', side=4, line=3)
```
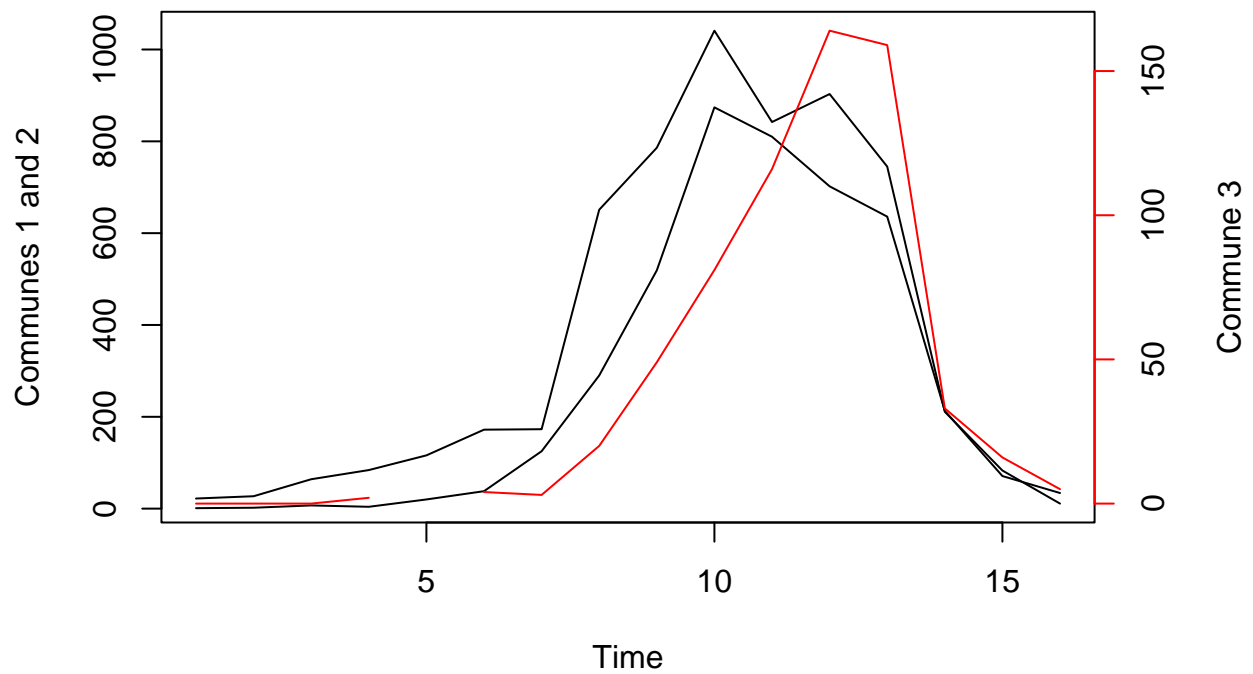
Exercise 4. Create a version of this figure with a stacked plot comparing Commune 3 with Communes 1 and 2.