

Births and Holidays

Data Computing

Computing project

The number of daily births in the US varies over the year and from day to day. What's surprising to many people is that the variation from one day to the next can be huge: some days have only about 80% as many births as others. Why?

The data table `Birthdays` in the `mosaicData` package gives the number of births recorded on each day of the year in each state from 1969 to 1988. (It would be nice to have more recent data, but I don't have them at hand.) For this activity, we'll work with data aggregated across the states.

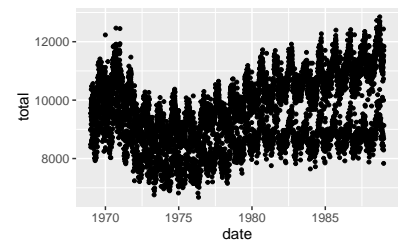
1. Create a new data table, `DailyBirths`, that adds up all the births for each day across all the states. Plot out daily births vs date.

The date variable in `Birthdays` prints out in the conventional, human-readable way. But it is actually in a format (called POSIX date format) that automatically respects the order of time. The `lubridate` package contains helpful functions that will extract various information about any date. Here are some you might find useful:

- `year()`
- `month()`
- `week()`
- `yday()` — gives the day of the year as a number 1-366. This is often called the "Julian day."
- `mday()` — gives the day of the month as a number 1-31
- `wday()` — gives the weekday (e.g. Monday, Tuesday, ...). Use the optional argument `label=TRUE` to have the weekday spelled out rather than given as a number 1-7.

Using these `lubridate` functions, you can easily look at the data in more detail.

2. To examine *seasonality* in birth rates, look at the number of births aggregated over all the years by
 - a. each week
 - b. each month
 - c. each Julian day
3. To examine patterns within the week, look at the number of births by day of the week.
4. Pick a two-year span of the `Birthdays` that falls in the 1980s, say, 1980/1981. Extract out the data just in this interval, calling it `MyTwoYears`. (Hint: `filter()`, `year()`). Plot out the births in this two-year span day by day. Color each date according to its day of the week. Explain the pattern that you see.



Births and holidays

- A few days each year don't follow the pattern in (4). We're going to examine the hypothesis that these are holidays. You can find a data set listing US federal holidays at <http://tiny.cc/dcf/US-Holidays.csv>.

Read it in as follows:¹

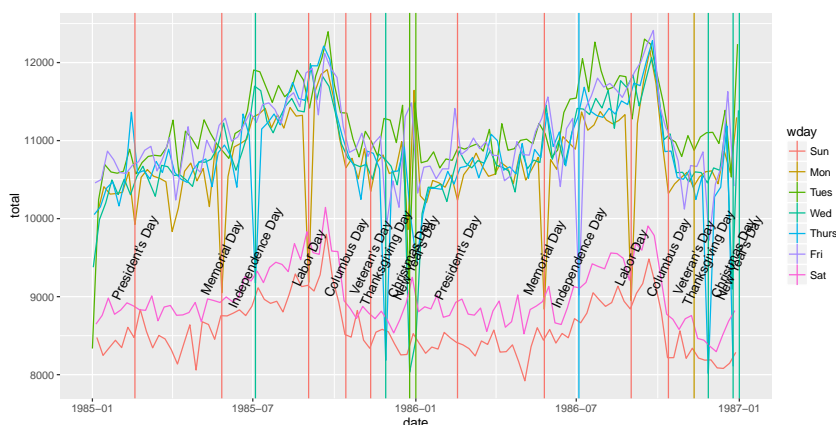
```
Holidays <- read.csv("http://tiny.cc/dcf/US-Holidays.csv") %>%
  mutate(date = as.POSIXct(lubridate::dmy(date)))
```

¹ The point of the `lubridate::dmy()` function is to convert the character-string date stored in the CSV to a POSIX date-number.

- Add a couple of layers to your plot from (4).
 - Draw a vertical bar at each date which is a holiday. You'll use the `geom_vline()` glyph. You can give a `data =` argument to `geom_vline()` to tell it to plot out the information from `Holidays` rather than `MyTwoYears`.²
 - Add a text label to each of the vertical bars to identify which holiday it is. Use the `geom_text()` glyph.³

² Unfortunately, due to what I believe is a bug in `geom_vline()`, you will have to set the x position of the bars by `as.numeric(date)` rather than just `date()`.

³ Hints: You'll have to make up a y-coordinate for each label. You can set the orientation of each label with the `angle` aesthetic.



- The plot in (6) is too busy. Let's explore some other ways to display the data to make it clearer to the view that holidays tend to be low-birth days.

Add a variable to `MyTwoYears` called `is_holiday`. It should be `TRUE` when the day is a holiday, and `FALSE` otherwise. One way to do this is with the transformation verb `%in%`, for instance,

```
is_holiday = date %in% Holidays$date
```

Make a new plot where you map `is_holiday` to the shape or the size aesthetics, or perhaps for faceting. Or perhaps change the way color is used in the graph to show weekends and holidays separately from non-holiday weekdays. Make the graph as simple as you can until you get one that clearly tells the story. You may want to simplify by eliminating any components of the graph (e.g. holiday labels? dots? lines? vertical lines?) that aren't essential to telling the story.