Holiday Birthdays

Stat 133

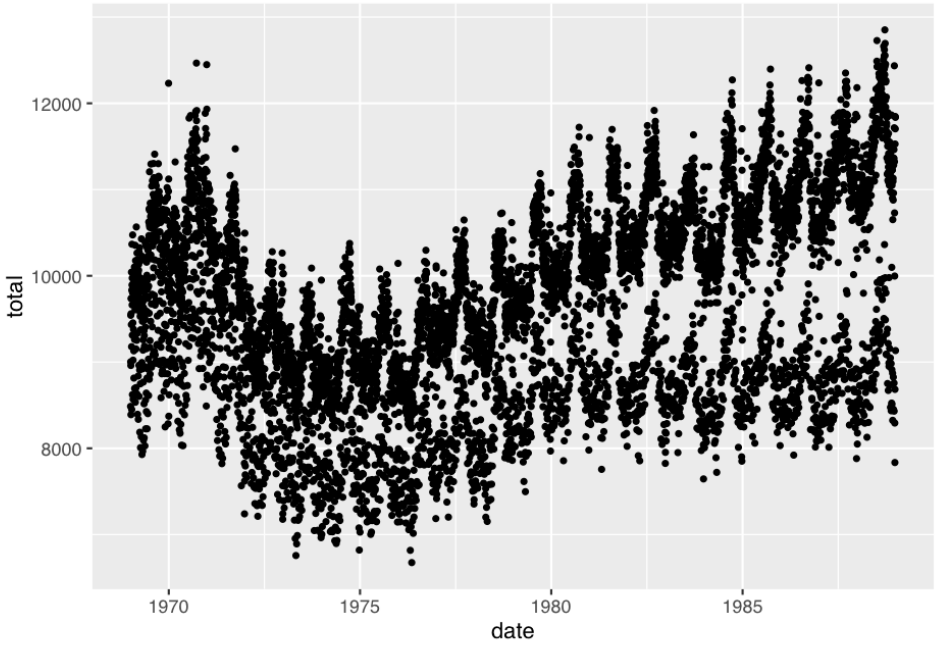
February 8, 2016

In this activity, you're going to examine how the number of daily births in the US varies over the years.

Births each day

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

5. 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 = lubridate::dmy(date))
```

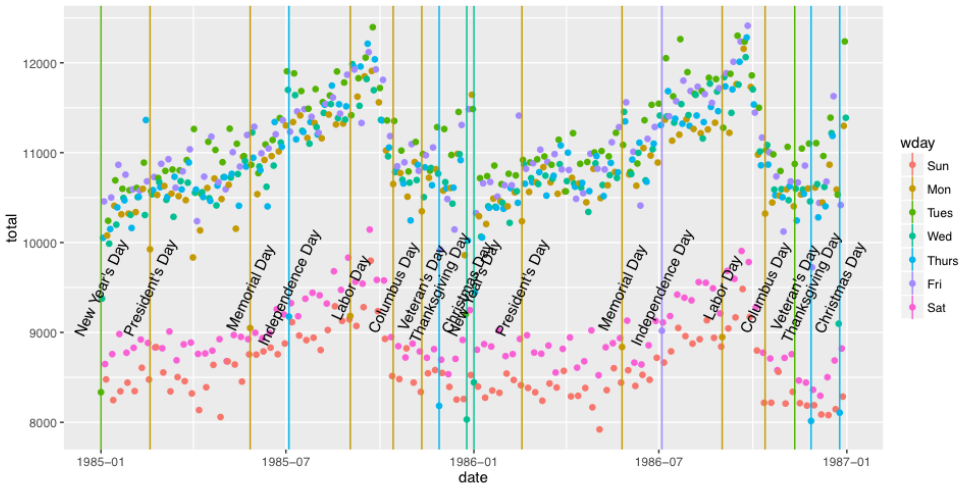
6. Add a couple of layers to your plot from (4).

    1. 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.[2]

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

[2] 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.



7. Join `MyTwoYears` with `Holidays`. You'll have to pick an appropriate form of join[3] to add a new column, `holiday` to every case in `MyTwoYears`. For those days that aren't a holiday, the correct choice of join function will create a value for `holiday` of `NA`.

8. Mutate the `holiday` variable to be "yes" or "no," depending on whether the day is a holiday or not. An appropriate argument to `mutate` would be
`is_holiday = ifelse(is.na(holiday), "no", "yes")`

9. Plot out the daily pattern over the two years of `MyTwoYears`, setting the *size* of the symbol to `is_holiday`. Is your hypothesis in (5) correct? If yes what holidays don't follow the pattern?

[3]Hint: `inner_join()`, `left_join()`, `anti_join()`?