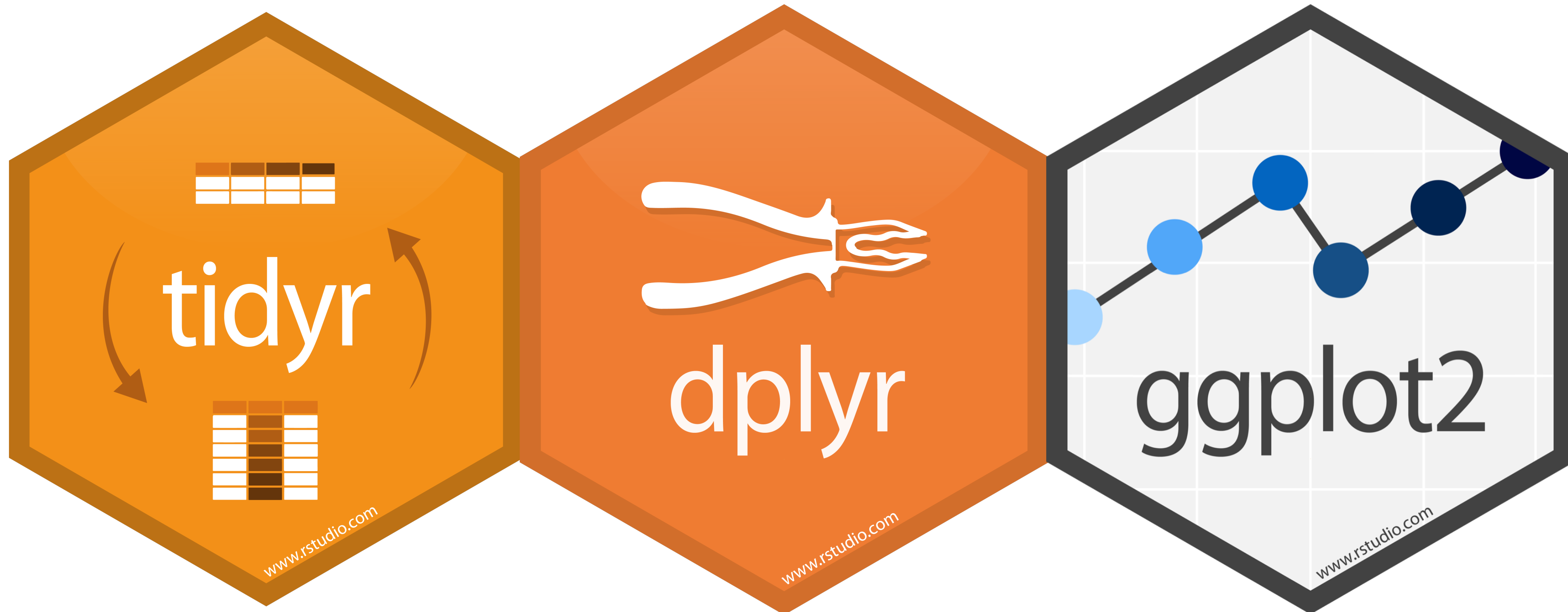


# Case Study





# Your Turn 0

1. Open 04-case-study.Rmd
2. Run the setup chunk

01:00



# fivethirtyeight

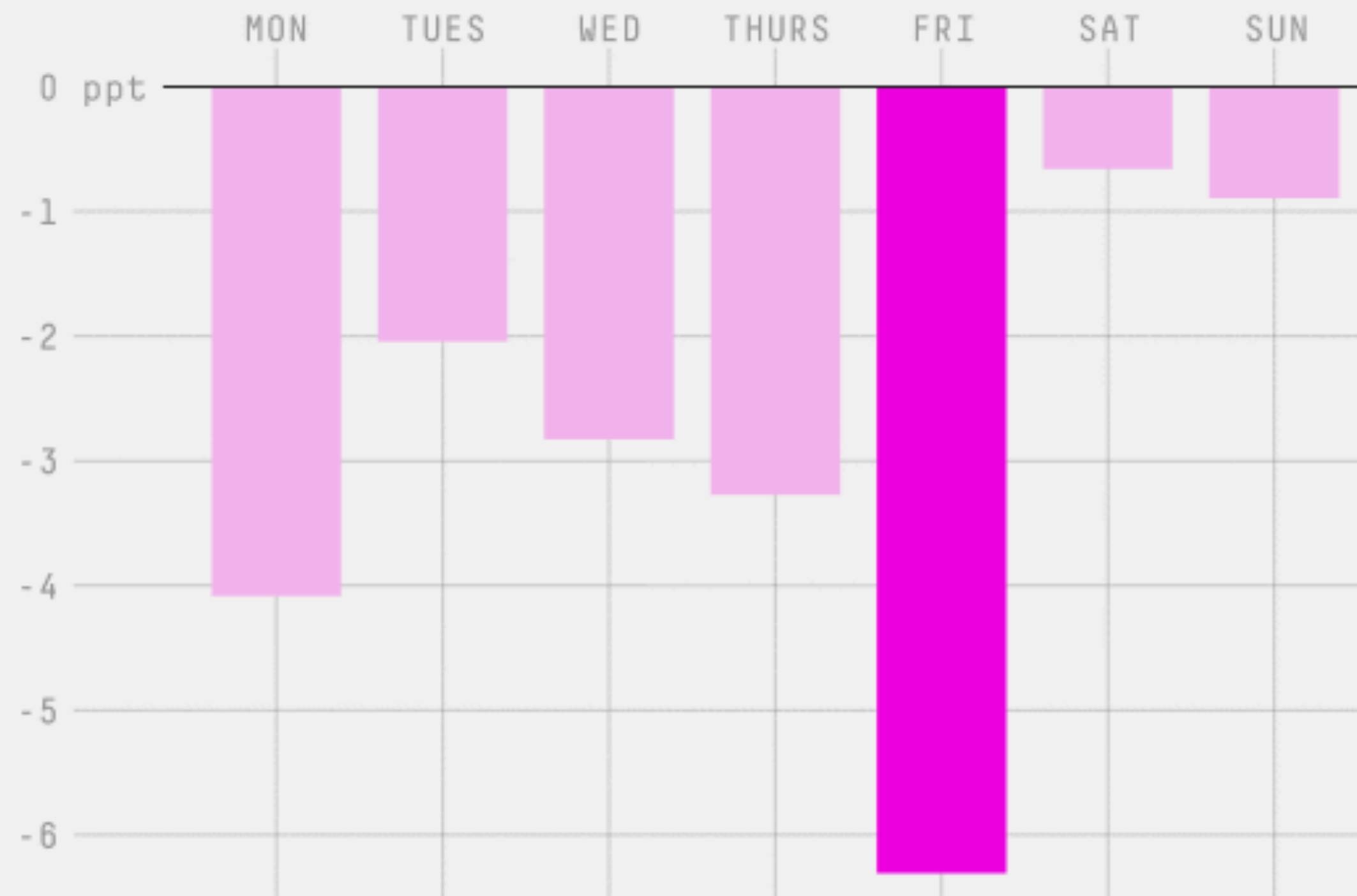


Datasets and code from the fivethirtyeight website.  
(Not officially published by 'FiveThirtyEight').

```
# install.packages("fivethirtyeight")  
library(fivethirtyeight)
```

## The Friday the 13th effect

Difference in the share of U.S. births on the 13th of each month from the average of births on the 6th and the 20th, 1994-2014



Excludes holidays

<https://fivethirtyeight.com/features/some-people-are-too-superstitious-to-have-a-baby-on-friday-the-13th/>

Can we  
replicate this  
plot?



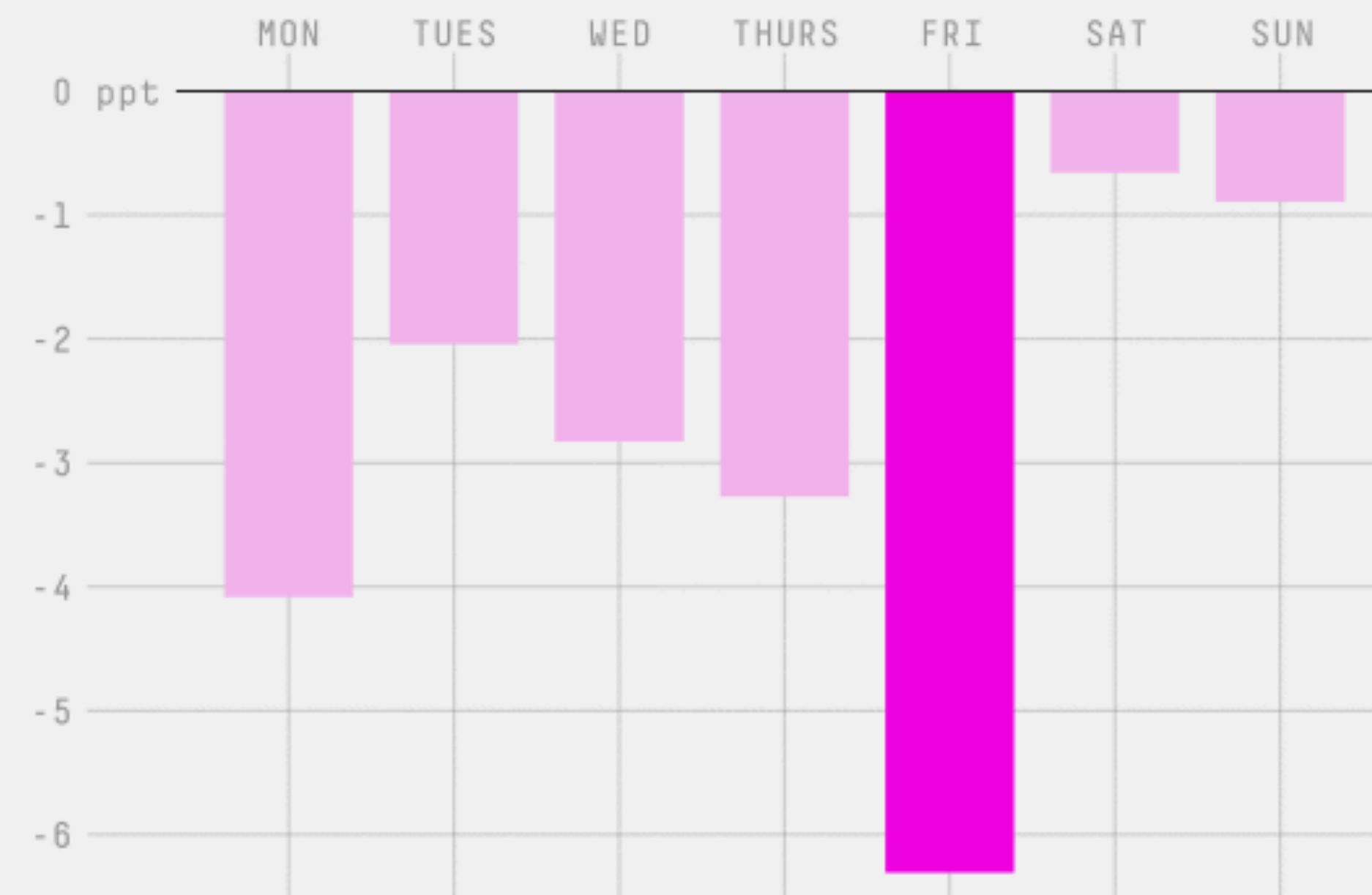
# Your Turn 1

Take a look at  
US\_births\_1994\_2003

With your neighbour,  
brainstorm the steps  
needed to get the data in  
a form ready to make the  
plot.

## The Friday the 13th effect

Difference in the share of U.S. births on the 13th of each month from the average of births on the 6th and the 20th, 1994-2014



Excludes holidays

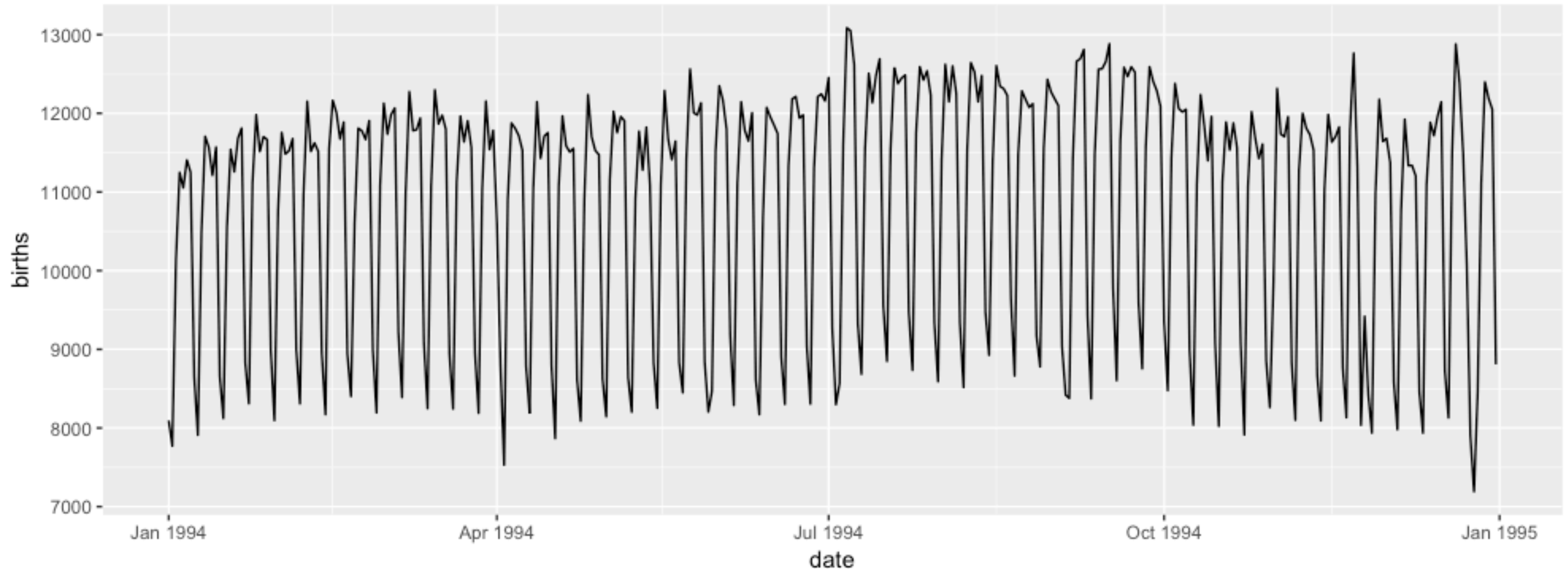
FIVETHIRTYEIGHT

SOURCES: CDC/NCHS, SOCIAL SECURITY ADMINISTRATION

05:00



```
US_births_1994_2003 %>%  
  filter(year == 1994) %>%  
  ggplot(mapping = aes(x = date, y = births)) +  
    geom_line()
```

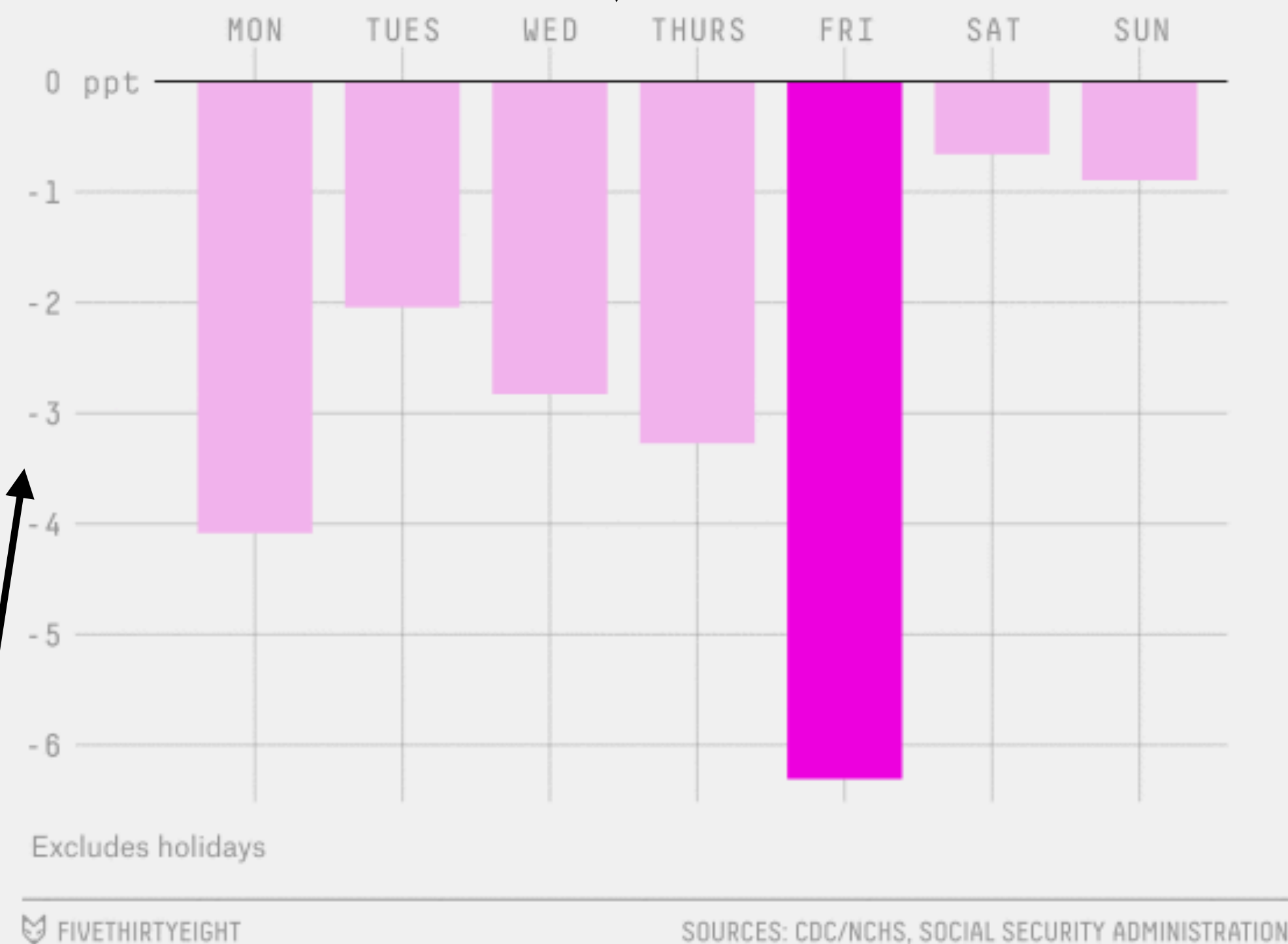


day\_of\_week

Data required to make the plot

### The Friday the 13th effect

Difference in the share of U.S. births on the 13th of each month from the average of births on the 6th and the 20th, 1994-2014



day\_of\_week

avg\_diff\_13\*

Mon

-2.69

Tue

-1.38

Wed

-3.27

...

...

\* using slightly different data

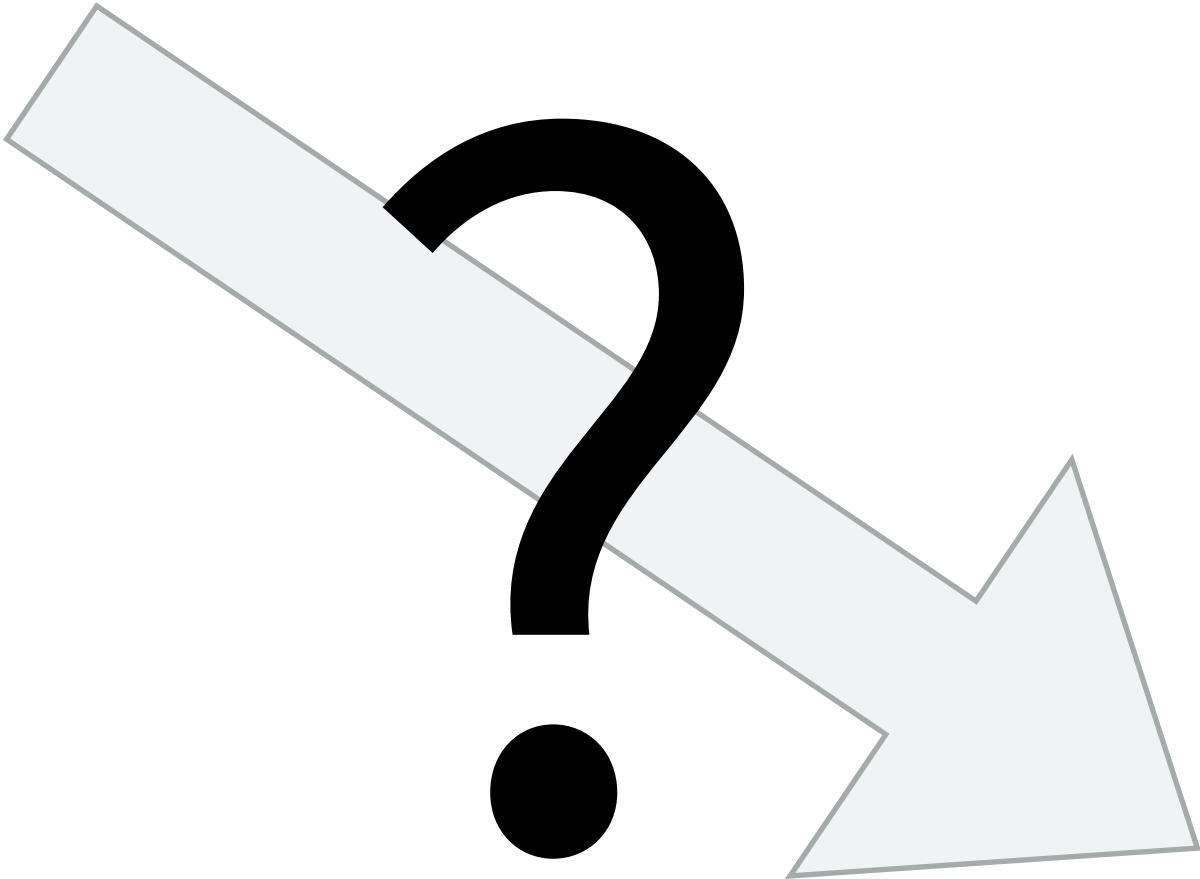
some calculated variable

# Start

# A tibble: 3,652 x 6						
	year	month	date_of_month	date	day_of_week	births
	<int>	<int>	<int>	<date>	<ord>	<int>
1	1994	1	1	1994-01-01	Sat	8096
2	1994	1	2	1994-01-02	Sun	7772
3	1994	1	3	1994-01-03	Mon	10142
4	1994	1	4	1994-01-04	Tues	11248
...						

?

?



# End

# A tibble: 7 x 2	
day_of_week	avg_diff_13
<ord>	<dbl>
1 Sun	-0.303
2 Mon	-2.69
3 Tues	-1.38
4 Wed	-3.27
5 Thurs	-3.01
6 Fri	-6.81
7 Sat	-0.738



# One such process

Get just the data for the 6th, 13th, and 20th

Calculate variable of interest:

(For each month/year):

Find average births on 6th and 20th

Find *percentage difference* between births on 13th and average births on 6th and 20th

Average *percent difference* by day of the week

Create plot



# Your Turn 2

Extract just the 6th, 13th and 20th of each month.

(`select(-date)` is removing the date column, because it gets in the way later and is redundant).

A digital timer display showing the time 02:00. The digits are black with a white outline, set against a white background. The timer is enclosed in a thin black rectangular border.



```
US_births_1994_2003 %>%  
  select(-date) %>%  
  filter(date_of_month %in% c(6, 13, 20))
```

year <int>	month <int>	date_of_month <int>	day_of_week <ord>	births <int>
1994	1	6	Thurs	11406
1994	1	13	Thurs	11212
1994	1	20	Thurs	11682
1994	2	6	Sun	8309
1994	2	13	Sun	8171
1994	2	20	Sun	8402
1994	3	6	Sun	8389
1994	3	13	Sun	8248
1994	3	20	Sun	8243
1994	4	6	Wed	11811

1-10 of 360 rows

Previous 1 2 3 4 5 6 ... 36 Next



# One month

Two options for arranging the data

## Option 1

days in rows

year <int>	month <int>	date_of_month <int>	day_of_week <ord>	births <int>
1994	1	6	Thurs	11406
1994	1	13	Thurs	11212
1994	1	20	Thurs	11682

## Option 2

days in cols

year <int>	month <int>	day_of_week <ord>	6 <int>	13 <int>	20 <int>
1994	1	Thurs	11406	11212	11682

Which one is  
tidy?



# Your Turn 3

Which arrangement is tidy?

(**Hint:** think about our next step *"Find the percent difference between the 13th and the average of the 6th and 12th"*. In which layout will this be easier using our tidy tools?)

02:00



Option 1

year	month	date_of_month	day_of_week	births
<int>	<int>	<int>	<ord>	<int>
1994	1	6	Thurs	11406
1994	1	13	Thurs	11212
1994	1	20	Thurs	11682

Next step, we'd have to write a custom function to summarize these three rows, relying on order, or subsetting to reference dates.  
NOT TIDY.

Option 2

year	month	day_of_week	6	13	20
<int>	<int>	<ord>	<int>	<int>	<int>
1994	1	Thurs	11406	11212	11682

Next step, we can use mutate directly referring to columns for days.  
TIDY!



# Your Turn 4

Tidy the filtered data to have the days in columns.

E.g.

	year <int>	month <int>	day_of_week <ord>	6 <int>	13 <int>	20 <int>
1	1994	1	Thurs	11406	11212	11682
2	1994	2	Sun	8309	8171	8402
3	1994	3	Sun	8389	8248	8243
4	1994	4	Wed	11811	11428	11585
5	1994	5	Fri	11904	11085	11645
6	1994	6	Mon	11130	10692	11337
7	1994	7	Wed	13086	12134	12378
8	1994	8	Sat	9336	9474	9646
9	1994	9	Tues	11448	12560	12584
10	1994	10	Thurs	12017	11398	11876

1-10 of 120 rows

Previous 1 2 3 4 5 6 ... 12 Next

03:00



```
US_births_1994_2003 %>%
  select(-date) %>%
  filter(date_of_month %in% c(6, 13, 20)) %>%
  pivot_wider(names_from = date_of_month, values_from = births)
```

	year <int>	month <int>	day_of_week <ord>	6 <int>	13 <int>	20 <int>
1	1994	1	Thurs	11406	11212	11682
2	1994	2	Sun	8309	8171	8402
3	1994	3	Sun	8389	8248	8243
4	1994	4	Wed	11811	11428	11585
5	1994	5	Fri	11904	11085	11645
6	1994	6	Mon	11130	10692	11337
7	1994	7	Wed	13086	12134	12378
8	1994	8	Sat	9336	9474	9646
9	1994	9	Tues	11448	12560	12584
10	1994	10	Thurs	12017	11398	11876

1-10 of 120 rows

Previous 1 2 3 4 5 6 ... 12 Next



# Your Turn 5

Now use `mutate()` to add columns for:

- The average of the births on the 6th and 20th
- The percentage difference between the number of births on the 13th and the average of the 6th and 20th

(Hint: You need to use backticks ``` around the days, e.g. ``6``, ``13`` and ``20`` to specify the column names)

03:00



```

US_births_1994_2003 %>%
  select(-date) %>%
  filter(date_of_month %in% c(6, 13, 20)) %>%
  pivot_wider(names_from = date_of_month, values_from = births) %>%
  mutate(
    avg_6_20 = (`6` + `20`)/2,
    diff_13 = (`13` - avg_6_20) / avg_6_20 * 100
  )

```

year <int>	month <int>	day_of_week <ord>	6 <int>	13 <int>	20 <int>	avg_6_20 <dbl>	diff_13 <dbl>
1994	1	Thurs	11406	11212	11682	11544.0	-2.87595288
1994	2	Sun	8309	8171	8402	8355.5	-2.20812638
1994	3	Sun	8389	8248	8243	8316.0	-0.81770082
1994	4	Wed	11811	11428	11585	11698.0	-2.30808685
1994	5	Fri	11904	11085	11645	11774.5	-5.85587498
1994	6	Mon	11130	10692	11337	11233.5	-4.82040326
1994	7	Wed	13086	12134	12378	12732.0	-4.69682689
1994	8	Sat	9336	9474	9646	9491.0	-0.17911706
1994	9	Tues	11448	12560	12584	12016.0	4.52729694
1994	10	Thurs	12017	11398	11876	11946.5	-4.59130289

1-10 of 120 rows

Previous 1 2 3 4 5 6 ... 12 Next



```

births_diff_13 <- US_births_1994_2003 %>%
  select(-date) %>%
  filter(date_of_month %in% c(6, 13, 20)) %>%
  pivot_wider(names_from = date_of_month, values_from = births) %>%
  mutate(
    avg_6_20 = (`6` + `20`)/2,
    diff_13 = (`13` - avg_6_20) / avg_6_20 * 100
  )

```

year <int>	month <int>	day_of_week <ord>	6 <int>	13 <int>	20 <int>	avg_6_20 <dbl>	diff_13 <dbl>
1994	1	Thurs	11406	11212	11682	11544.0	-2.87595288
1994	2	Sun	8309	8171	8402	8355.5	-2.20812638
1994	3	Sun	8389	8248	8243	8316.0	-0.81770082
1994	4	Wed	11811	11428	11585	11698.0	-2.30808685
1994	5	Fri	11904	11085	11645	11774.5	-5.85587498
1994	6	Mon	11130	10692	11337	11233.5	-4.82040326
1994	7	Wed	13086	12134	12378	12732.0	-4.69682689
1994	8	Sat	9336	9474	9646	9491.0	-0.17911706
1994	9	Tues	11448	12560	12584	12016.0	4.52729694
1994	10	Thurs	12017	11398	11876	11946.5	-4.59130289

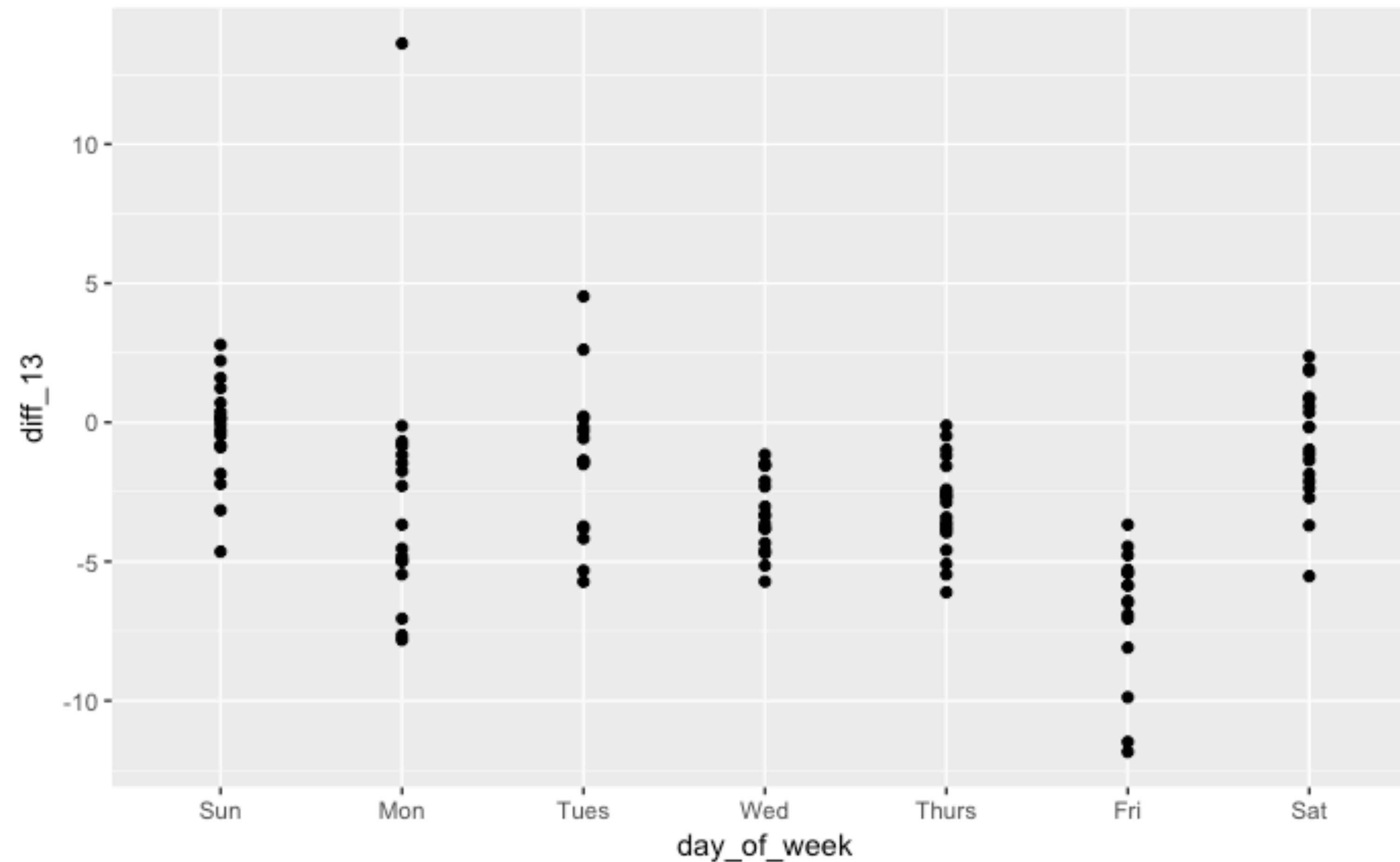
1-10 of 120 rows

Previous 1 2 3 4 5 6 ... 12 Next



```
births_diff_13 %>%
```

```
  ggplot(mapping = aes(day_of_week, diff_13)) +  
    geom_point()
```





```
births_diff_13 %>%  
  filter(day_of_week == "Mon", diff_13 > 10)
```

year	month	day_of_week	6	13	20	avg_6_20	diff_13
<int>	<int>	<ord>	<int>	<int>	<int>	<dbl>	<dbl>
1999	9	Mon	8249	11481	11961	10105	13.61702

1 row



# Your Turn 6

Summarize each day of the week to have mean of `diff_13`.

Then, recreate the fivethirtyeight plot. (**Hint:** if you specify a y aesthetic with `geom_bar()` you'll need to add `stat = "identity"` as an argument.

(**Extra challenge:** use a different summary, and/or another way of visualizing the data)



05:00



```

US_births_1994_2003 %>%
  select(-date) %>%
  filter(date_of_month %in% c(6, 13, 20)) %>%
  pivot_wider(names_from = date_of_month, values_from = births) %>%
  mutate(
    avg_6_20 = (`6` + `20`)/2,
    diff_13 = (`13` - avg_6_20) / avg_6_20 * 100
  ) %>%
  group_by(day_of_week) %>%
  summarise(avg_diff_13 = mean(diff_13))

```

day_of_week <ord>	avg_diff_13 <dbl>
Sun	-0.3026934
Mon	-2.6856859
Tues	-1.3776517
Wed	-3.2735133
Thurs	-3.0117652
Fri	-6.8057874
Sat	-0.7376400



```

births_13_sum <- US_births_1994_2003 %>%
  select(-date) %>%
  filter(date_of_month %in% c(6, 13, 20)) %>%
  pivot_wider(names_from = date_of_month, values_from = births) %>%
  mutate(
    avg_6_20 = (`6` + `20`)/2,
    diff_13 = (`13` - avg_6_20) / avg_6_20 * 100
  ) %>%
  group_by(day_of_week) %>%
  summarise(avg_diff_13 = mean(diff_13))

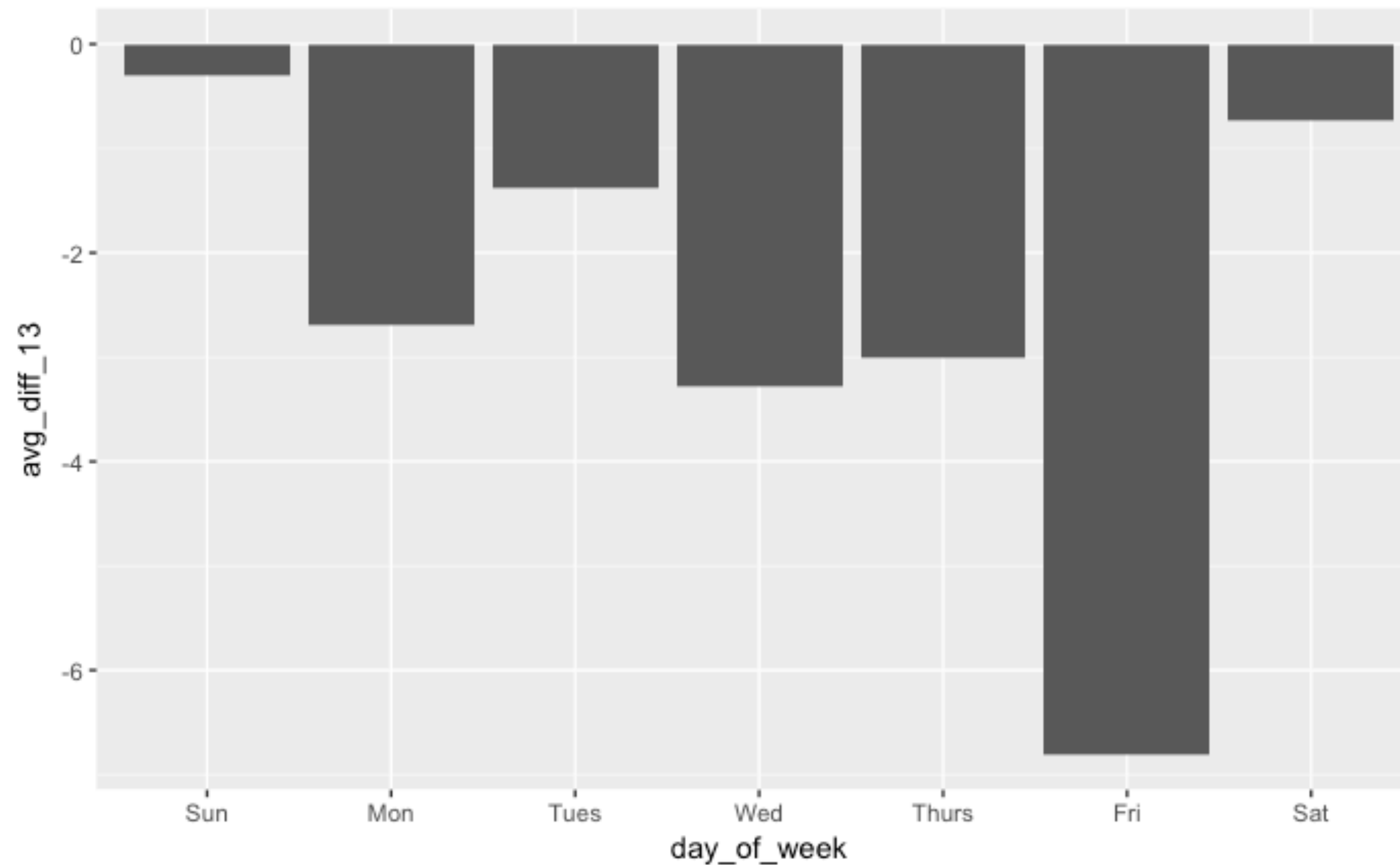
```

<b>day_of_week</b> <ord>	<b>avg_diff_13</b> <dbl>
Sun	-0.3026934
Mon	-2.6856859
Tues	-1.3776517
Wed	-3.2735133
Thurs	-3.0117652
Fri	-6.8057874
Sat	-0.7376400



```
births_13_sum %>%
```

```
  ggplot(aes(x = day_of_week, y = avg_diff_13)) +  
    geom_bar(stat = "identity")
```





# Extra Challenges

If you wanted to use the `US_births_2000_2014` data instead, what would you need to change in the pipeline? How about using both `US_births_1994_2003` **and** `US_births_2000_2014`?

Try not removing the `date` column. At what point in the pipeline does it cause problems? Why?

Can you come up with an alternative way to investigate the Friday the 13th effect? Try it out!



# Case Study

