

# Exploratory data analysis

Week 4, Lecture 8

<https://live.ds306.org>

September 24, 2026

# From Lecture 7

---

- We learned about exploratory data analysis.
- We learned which types of plots are appropriate for different types of data.
- We learned how to interpret patterns and anomalies in data distributions.

We will use this same process on new datasets in the following examples.

# Do runners aim for particular finishing times?

---

What would you expect a histogram of marathon finishing times to look like?

# One row per marathon finish

---

```
marathon |> head()
```

- `finish_minutes` measures time from crossing the start line to crossing the finish line.
- `marathon` identifies the race; `year` identifies its year.
- One row is a finishing time for one runner. The same runner could appear in several races.

Source: [Allen, Dechow, Pope, and Wu: marathon data](#).

# The distribution of marathon times

---

```
hist(marathon$finish_minutes)
```

What does this histogram tell you about the distribution of marathon finishing times?

# Could these bins hide a pattern?

---

Suppose many runners aim to finish in under four hours (240 minutes). Where would you look for evidence of this? How would this pattern show up in a histogram?

# Inspect the times near four hours

Zoom in on marathon times near four hours. Plot the resulting distribution.

(To draw a vertical reference line on the plot, add `geom_vline(xintercept = 240)` to your plot).

```
marathon
```

- Filter the data to finishing times that are within ten minutes of four hours.
- Plot a histogram of `finish_minutes`.
- Label the axes appropriately.

# What appears with one-minute bins?

---

```
marathon |>
  filter(finish_minutes >= 230, finish_minutes < 250) |>
  ggplot(aes(x = finish_minutes)) +
  geom_vline(xintercept = 240, color = "#b5422d", linewidth = 1) +
  labs(x = "Finishing time (minutes)", y = "Number of finishes") +

  geom_histogram(binwidth = 1, boundary = 0, closed = "left", fill = "#2364aa", color = "white")
```

Compare the bars immediately before and after the line. How did the first histogram combine these observations?



# Count finishes on each side

---

Compare equal-length intervals:  $[239, 240)$  and  $[240, 241)$ . In `sum(condition)`, each TRUE contributes 1 and each FALSE contributes 0.

```
marathon |>
  summarize(
    just_under_4h = sum(finish_minutes >= 239 & finish_minutes < 240),
    just_over_4h = sum(finish_minutes >= 240 & finish_minutes < 241)
  )
```

Exactly 240 minutes belongs to the second interval. These boundaries match the histogram's `closed = "left"` setting.

# Make a hypothesis

---

Formulate a hypothesis about why there are more finishes just under four hours than just over four hours.

# Is there a general pattern?

If I subtract the number of finishes in the previous minute from the number of finishes in the current minute, what do I expect to see?

```
marathon |>
  filter(finish_minutes >= 150,
         finish_minutes < 391)
```

- Add column `minute` by rounding `finish_minutes` down with `floor()`.
- Count the finishes in each minute.
- Arrange the rows in increasing order of `minute`.
- Create `change = n - lag(n)`: this minute's count minus the previous minute's count.
- Make a plot of `change` against `minute`, with informative axis labels.

# Communicate your finding

---

Summarize the findings of your data exploration in a few sentences, highlighting the patterns you observed and any potential explanations for these patterns. Is there any limitation to your analysis?

# Are birthdays uniformly distributed?

---

Are birthdays equally likely to occur on every day? Month? Year?

# Birth data

---

```
births |> head()
```

- Daily U.S. birth counts for 2000–2014; 5,479 rows.
- `year`, `month`, and `date_of_month` identify the date.
- `day_of_week` is a factor with levels Sun, Mon, Tue, Wed, Thu, Fri, Sat.
- `births` counts births on that date.

Source: [SSA data published by FiveThirtyEight](#).

# Generate a hypothesis

---

What pattern do you expect to see if we plot the number of births for each day of the week? Why do you expect that pattern?

# Visualize births per day of the week

```
births |> head()
```

- Group births by day\_of\_week.
- Calculate mean\_births, the average of births for each weekday.
- Save the seven-row result as weekday\_births.
- Use geom\_col() to plot one bar per weekday, with height given by



# Weekdays or holidays?

---

July 4 falls on a different weekday each year.

What would you expect July 4 births to look like across years if births are less common on certain weekdays? What would the pattern look like if births were less common on holidays as well as weekends?

# July 4 births across years

---

births

- Filter `births` to July 4 in every year and save the result.
- Make a bar plot with `year` on the x-axis and `births` on the y-axis.
- Color each bar by its `day_of_week`.
- Label the axes and give the plot a title.

# Would you choose a memorable birthday?

---

Think about dates such as 07/07/07 or 12/12/12.

Would you expect more or fewer births on dates like these? What comparison would show whether relatively more or fewer births occur on these dates?

# Which days should we compare?

---

For each repeated date, compare births with **one week before** and **one week after**. All three dates fall on the same weekday.

For July 7, 2007, we need:

| One week before | Repeated date | One week after |
|-----------------|---------------|----------------|
| June 30, 2007   | July 7, 2007  | July 14, 2007  |

The data contain every day. Once sorted by year, month, and day, moving seven rows takes us one week.

# Look seven rows backward and forward

---

We used `lag()` to get the previous value in the marathon data. `lag(births, 7)` gets the value seven rows earlier; `lead(births, 7)` gets the value seven rows later.

```
births |>
  arrange(year, month, date_of_month) |>
  mutate(before = lag(births, 7),
         after = lead(births, 7)) |>
  filter(year == 2007, month == 7, date_of_month == 7) |>
  select(year, month, date_of_month, before, births, after)
```

# Compare all twelve repeated dates

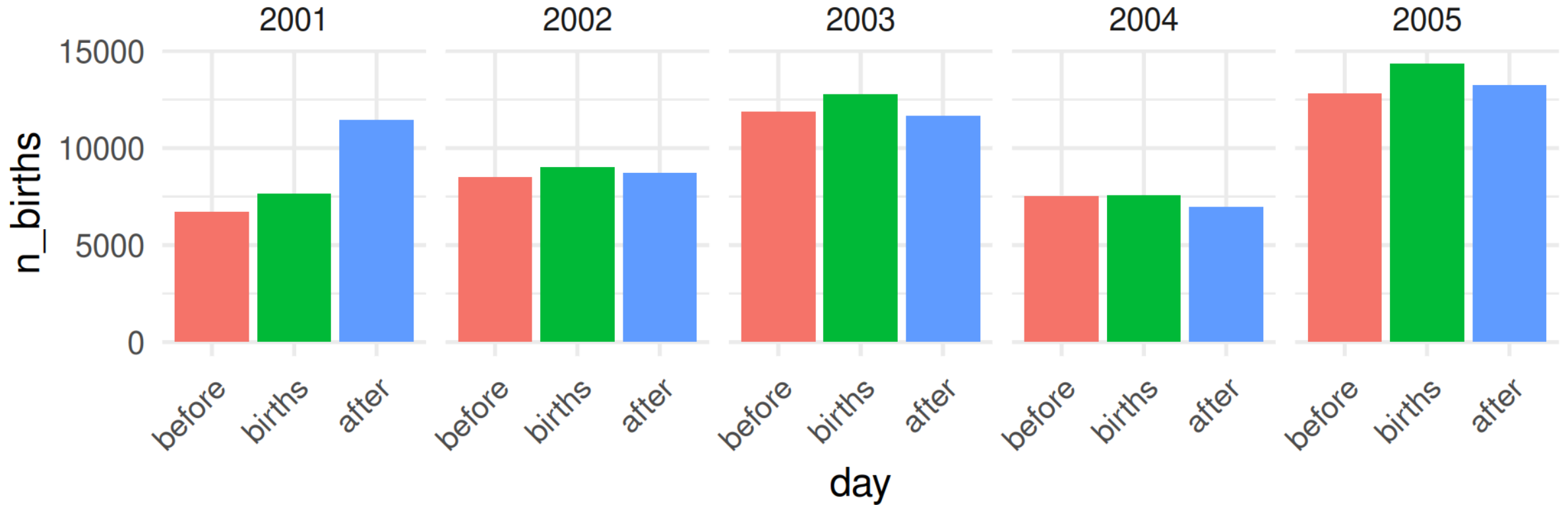
For 01/01/2001 through 12/12/2012, month, date\_of\_month, and year - 2000 match. Build a table with the birth count and its two neighboring weekly counts.

```
births |> head()
```

- Sort births by year, month, and date\_of\_month.
- Before filtering, create before = lag(births, 7) and after = lead(births, 7).
- Keep rows where month, date\_of\_month, and year - 2000 match; save as repeat\_comparison.
- Display year, month, date\_of\_month, before, births, and after.

# The plot we want

For each year, compare the repeated date with one week before and one week after.



# Work backward from the plot

---

What information does R need to draw each bar?

| Part of the plot                      | Column we need |
|---------------------------------------|----------------|
| Which panel the bar belongs to        | year           |
| How tall the bar is                   | n_births       |
| Which of the three days it represents | day            |



# Table needed for plot

---

For 2007, our current table has three count columns:

```
repeat_comparison |>
  filter(year == 2007) |>
  select(year, before, births, after)
```

The plotting table needs three rows for 2007:

```
repeat_long |>
  filter(year == 2007) |>
  select(year, day, n_births)
```

The column names become values in day. Their counts go into n\_births.

# Plan the reshape

---

Use `pivot_longer()`, as we did with the movie data, to build the plotting table.

```
repeat_comparison
```

- Keep year and the three counts: before, births, and after.
- Gather the three count columns into `n_births`, recording their names in `day`.
- Keep the year attached to each count: three rows per year, 36 rows total. Save the table as `repeat_long`.

# Plot the three counts

---

Use the same bar-plot commands we have used before, then add one panel per year.

```
repeat_long |>
  mutate(day = factor(
    day,
    levels = c("before", "births", "after")))
ggplot()
```

- Map `day` to `x`, `n_births` to `y`, and `day` to `fill`.
- Add bars with `geom_col()`.
- Add `facet_wrap(~ year)` to give each year its own panel.

# Births around repeated dates

```
repeat_polished <- repeat_long |>
  mutate(day = factor(day, levels = c("before", "births", "after"))) |>
  ggplot(aes(x = day, y = n_births, fill = day)) +
  geom_col(width = 0.7) +
  facet_wrap(~ year, ncol = 4, nrow = 3) +
  scale_fill_manual(
    values = c(before = "#BCC5CE", births = "#0072B2", after = "#BCC5CE")) +
  scale_x_discrete(labels = c(before = "Before", births = "Repeated", after = "After")) +
  scale_y_continuous(labels = scales::label_comma()) +
  labs(x = NULL, y = "Number of births",
       subtitle = "One week before · Repeated date · One week after") +
  theme_minimal(base_size = 16) +
  theme(legend.position = "none",
        panel.grid.major.x = element_blank(),
        panel.grid.minor = element_blank(),
        strip.text = element_text(face = "bold"),
        axis.text.x = element_text(size = 12))
repeat_polished
```

# Communicate your findings

---

What do these plots tell us? What patterns did you find? Did you notice any exceptions or unusual dates?

# **Was the Vietnam draft fair?**

---

# What was the draft?

---

During the Vietnam War, the U.S. government required selected young men to enter military service. This compulsory service was called **the draft**. Being drafted would interrupt school, work, and family plans, and military service in Vietnam was dangerous for those who served.

# Casualties in the Vietnam War

`vietnam_casualties`

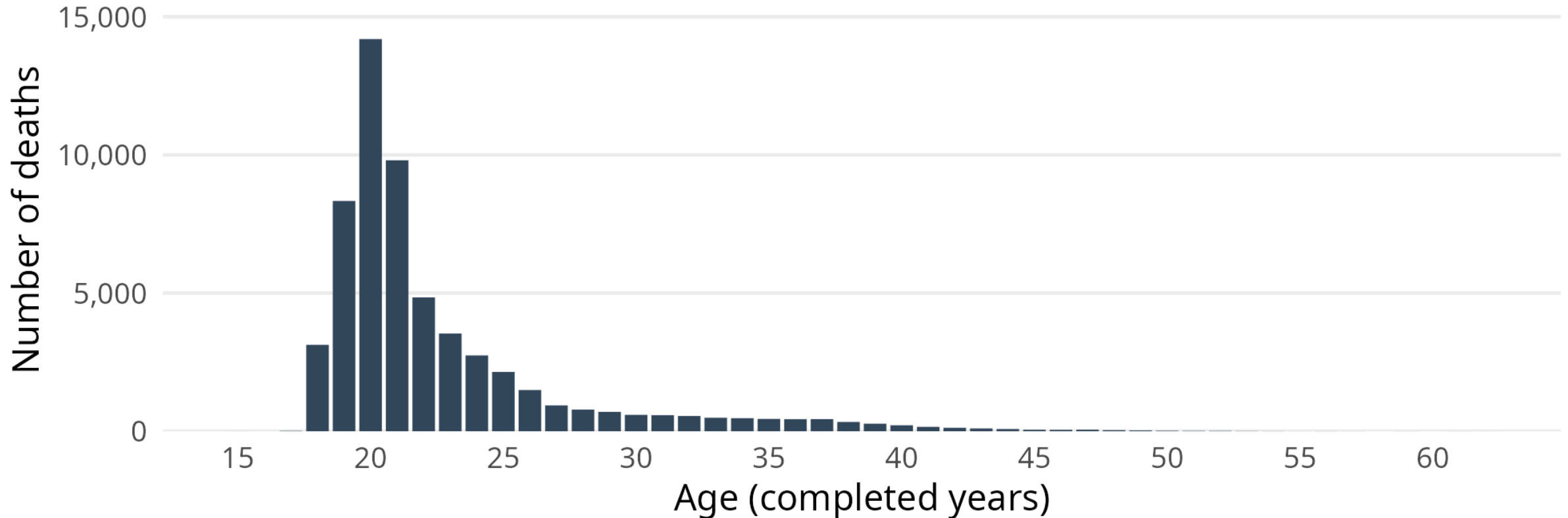
- Starting with all rows of `vietnam_casualties`, count the number of people at each age.
- Map age to x and your count column to y.
- Use `geom_col()` to draw one bar per age.
- Label the axes and title the plot to identify U.S. military deaths in the Vietnam War



# Ages of those who died

## Ages of U.S. military personnel who died in the Vietnam War

58,220 recorded deaths · Median age: 21



Source: National Archives, DCAS Vietnam extract (April 29, 2008).  
Age calculated from birth date and recorded incident/death date.

Bar plot of the ages of all 58,220 U.S. military fatal casualties in the National Archives Vietnam extract.

# How the draft worked

---

On **December 1, 1969**, officials drew 366 birthday capsules to set the 1970 call-up order for men born in 1944–1950.

The first birthday drawn was **September 14**: lottery number **1**.

A low number meant earlier call-up. The intended procedure gave every birthday the same chance of being drawn early.



Alexander Pirnie draws a capsule at the December 1, 1969 draft lottery.

Selective Service photograph, December 1, 1969. Public domain.

# The draft

---

CBS News live coverage, December 1, 1969.



**Video player configuration error**

Error 153

[Watch video on YouTube](#)

[Learn more](#)

Error 153

[Open the broadcast on YouTube](#)

# Was the draft fair?

---

In your opinion, what does it mean for the draft to be fair?

# Plan the audit

---

If priority is unrelated to birthday, we would not expect any systematic pattern across the calendar.

```
draft |> head()
```

- Make a scatterplot of Rank against Day using all birthdays in draft.
- Group by Month and calculate mean\_rank, the mean of Rank, and n\_dates, the number of birthdays.

# Look across the calendar

---

Adding a line of best fit (regression line):

```
draft |>
  ggplot(aes(x = Day, y = Rank)) +
  geom_point(alpha = 0.6, color = "#2364aa") +
  labs(x = "Birthday: day of year", y = "Draft priority (lower = earlier)",
       caption = "1969 drawing for the 1970 draft; vcdExtra::Draft1970") +
  geom_smooth(method = "lm", formula = y ~ x, se = FALSE, color = "#D55E00")
```

# Monthly pattern

Compare the distribution of lottery numbers across birth months. Use a boxplot to show the spread within each month.

```
draft |> head()
```

- Use all birthdays in draft, mapping Month to x and Rank to y.
- Add `geom_boxplot()` to make one boxplot per month, ordered January through December.
- Label the axes so readers know that smaller lottery numbers mean earlier call-up.

# Interpret the monthly pattern

What was the average lottery number for the first six months? What about the last six? Was the difference statistically significant?

The starter marks January–June with `first_half = TRUE` and July–December with `FALSE`. Use `t.test(Rank ~ first_half, data = draft_halves)` to compare the two means.

```
draft_halves <- draft |>
  mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun"))

draft_halves |> head()
```

- Calculate the mean Rank for each half of the year using `group_by()` and `summarize()`.
- Run a two-sample t-test comparing Rank between the two halves of the year.
- In a code comment, report both means and explain whether the difference is significant at the 5% level and which half faced earlier call-up.



# Why did this happen?

---

1. Each capsule contained a birthday. **January went in first; December went in last.**
2. The capsules were not mixed enough to undo that ordering. Late-year birthdays were drawn disproportionately early.
3. **The draw order became the lottery number:** first drawn = 1, second = 2, and so on.
4. Men with lower numbers were called up sooner. **Late-year birthdays therefore faced a greater chance of being drafted.**

For a fair lottery, every birthday needed the same chance of being drawn early. Insufficient mixing let birth month influence who was called up first.

Source: [Starr, \*Nonrandom Risk: The 1970 Draft Lottery\*](#).

# Communicate the lottery finding

---

Explain who was disadvantaged by the lottery and how the mixing procedure produced the pattern. What pattern in the plots supports your conclusion?

# What did the follow-up comparisons change?

| Case          | First comparison                   | Follow-up                                          |
|---------------|------------------------------------|----------------------------------------------------|
| Marathons     | Broad histogram of finishing times | One-minute bins and counts around target times     |
| Birthdays     | Births by weekday                  | July 4 and repeated dates vs. nearby same weekdays |
| Draft lottery | Lottery numbers                    | Lottery number by birthday and month               |

For each case, identify the new information used in the follow-up.

# Sources

---

- [Daily U.S. births: SSA via FiveThirtyEight](#), 2000–2014.
- [Marathon results: Allen, Dechow, Pope, and Wu](#), 2003–2013; see the [classroom sample documentation](#).
- [1970 draft data and historical references: vcdExtra](#).

# Week 4, Lecture 8: Three data investigations

Closed September 24, 2026 at 3:45 PM

21 class questions · 1276 anonymous responses · 0 student questions

The following slides preserve what students submitted during class. No names or login information are included.

# What does this histogram tell you about the distribution of marathon finishing times?

---

79 anonymous responses

|                                                                                                                                                                     |           |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| <b>Most finish around 200–300 minutes</b>                                                                                                                           | <b>36</b> |
| <small>States that the majority of runners complete the marathon in the 200–300 minute range (typical, central concentration).</small>                              |           |
| <b>Distribution is right-skewed</b>                                                                                                                                 | <b>31</b> |
| <small>Identifies the distribution as skewed to the right, indicating a longer tail toward slower (larger) times.</small>                                           |           |
| <b>Mentions center, mean, or median values</b>                                                                                                                      | <b>7</b>  |
| <small>Comments estimating the distribution's center, mean, or median (e.g., around 250–275 minutes) or notes range and modality alongside a central value.</small> |           |
| <b>Right skew implies mean greater than median</b>                                                                                                                  | <b>3</b>  |
| <small>Notes that because the distribution is skewed to the right, the mean is larger than the median (explicitly relating skew to summary statistics).</small>     |           |

# Response themes for question 1

---

Other responses

---

2

# Student responses

---

1. A huge percentage of the people fit between 200-300, it looks like a standard histogram with the middle being the most popular and the ends going down from the middle.
2. Finish minutes tend to be in the 200-300 minutes.
3. It is slightly skewed to the right but there seems to be a higher frequency of finishing times between 250 to 300 minutes
4. majority of racers finished the marathon between 200 and 300 minutes and it is right skewed
5. Marathon finish times have an average of about 300 minutes and its skewed to the right.
6. Marathon finishing times are mostly centered around 250-300 minutes, with the frequency decreasing as the marathon times increase. It is skewed to the right.
7. More concentrated in the middle, with few in finishing really quickly and finishing slowly.
8. most finish in 250 min
9. Most finish times cluster around 200-300 minutes. So the average must be somewhere between 250-300 minutes



# Student responses

---

1. most of them are in 200-300
2. most people finish between 200-300 minutes; skewed right
3. Most people finish between 200-300 minutes.
4. most people finish in around 275 minutes, with more people taking longer than the majority than there are faster marathon runners
5. Most people finish marathons in 300 minutes
6. Most people finish running a marathon between 200 and 300 minutes. The range goes from 100 to 700. It is right skewed.
7. Most people finished a marathon betterwn 200-300 minutes
8. most people finished in 300min
9. Most people take between 200-300 minutes to run a marathon, with a few people taking much longer.

# Student responses

---

1. most runner finish in 200-300 min
2. Most runners finish between 200 and 350 minutes (about 3–6 hours), with the highest number finishing around 250–300 minutes. The distribution is right-skewed, meaning fewer runners take much longer to finish.
3. Most runners finish between 200 minutes and 400 minutes, with most frequent time being around 250-300
4. most runners took about 200-300 minutes to finish, and the distribution is not completely even, it is skewed right.
5. normal
6. Seems relatively bell curved, most people finish in ~250-300 minutes, the fastest times were recorded between 100-150 minutes
7. That most runners usually finish between 250 to 300 minutes
8. The distribution is unimodal. The mean finishing time is about 270 minutes.
9. The histogram tells me that there are more skewed left finish minutes. Most people finish around 200-300 minutes for a marathon

# Student responses

---

1. the histogram tells us that the distribution is centered around 300 minutes with a small right skew.
2. The majority of finishing time in minutes is between 200 and 300 minutes, with all values being between 100 and 700 minutes. It is right skewed, but most are generally centrally clustered.
3. The marathon finishing times are centered around 250-300 minutes, with a right-sided tail formed of less common longer times.
4. The most frequent finish time is about 275 mins, more than half are finished between 200 and 300 mins.
5. The most common finish times is in the 250-300 min range. It looks like the median is also around that area, maybe in the 200-250 min group.
6. The most common finishing times are just under 300 minutes, and the values center around this value. It also has a right tail.
7. This histogram shows that the most frequent marathon times fall somewhere between 200 and 300 minutes; it's bell shaped with a right skew.
8. This tells me that the majority of marathon finishers finish the marathon in about 200-350 minutes, with the peak being from 250-300. The very few finish later than that.

# Student responses

---

1. which is most frequent and which how are those distributed

# Student responses

---

1. Centered around 275 minutes, right skewed
2. it is a relatively normal distribution that is skewed right a little
3. It is a right skew curve.
4. It is skewed right, so the times are clustered around 250, but there are some finishers who take around 700
5. most of the people finish the race in a period of 200-350 minutes, ang the distribution of the finishing minutes has a long right tail
6. right skewed
7. Right skewed
8. right skewed. maybe because the people who choose to run marathons tend to be better runners. selection bias
9. severe dropoff after 300(5hrs)

# Student responses

---

1. skewed slightly right, unimodal
2. Slightly skewed right, tells us that the most common finishing time is around 250-300 minutes.
3. That the distribution is right skewed and normal.
4. the data is close to being normally distributed with a slight skewness to the right. the median seems to be just under 300 minutes.
5. The data is skewed right, meaning that the mean should be greater than the median due to more varying / outlier times with greater values.
6. The distribution is right-skewed.
7. The distribution is skewed to the right and looks like a gamma distribution. The meet finish seems to be around 200-300 minutes
8. The distribution is unimodal and slighted skewed right. It centers at around 270 minutes and has a range of 500 minutes.
9. The distribution is unimodal with a peak around 250 minutes. The median of the data likely lies between 200 and 300 minutes.

# Student responses

---

1. The graph is right skewed, where the most people finished around 250 mins.
2. The histogram has a right skew, which means that the data is centralized to the left of the graph. It appears that the majority of people finish in the 200-350 minute range, with some scattered before and a few people stretching into the hours after
3. The histogram is right-tailed distribuion, and the mean is centered around 270 mins. This indicates that indeed people tend to have a centralized finishing time
4. The histogram is skewed right. People finished marathons between 200-300 minutes most frequently.
5. The histogram is unimodal and skewed to the right. We can see that marathon times range btween 100 to 600 minutes, with a peak between 250 to 300 minutes in finish time.
6. The histogram shows us that there is a right hand skew to the data. We see that most of the data lives toward the left hand side, meaning more people were finishing in less time. This make sense because some marathons require you to be able to finish within a certain time frame

# Student responses

---

1. The marathon finish time between 250 and 300 minutes has the highest frequency, and the data has a right skew, meaning there are some higher finishing times causing the mean to be higher than the median.
2. There is a moderate right skew, the mean minutes finishes is around 280.
3. There is an average run time of around 250 minutes, slightly right skewed.
4. This histogram is a unimodal skewed right distribution with a center around 275.
5. This histogram looks like it is unimodal with a right skew. It seems to be centered around 275 minutes but again is right skewed, so some people finished in much slower times that may bring the average up
6. Unimodal, somewhat right skewed, mean around 250 minutes
7. We see that the histogram of finish times is slightly right skewed, with a peak around 250ish mins



# Student responses

---

1. It tells us about where the average time will be and the distribution of times for marathon runners centered around 250 minutes
2. The histogram shows that the average time is around 300 mins or 5 hours and is right skewed towards higher times.
3. There is a range of about 10 hours in which it can take someone to complete a marathon. The mean is likely around 250 minutes, with a unimodal distribution.
4. This tells use that the median time to finish is somewhere between 250 to 300 minutes. It is also skewed to a lower number of minutes to finish.
5. Unimodal
6. we know the mean run time is somewhere between 250 and 300 roughly and very denser in the 200 to 250 range
7. We see a strong right tail and a single peak, showing a group of runners outside the around-mean grouping.

# Student responses

---

1. Its skewed to the right, which means the mean > median.
2. The data is slightly skewed to the right, meaning the mean should be bigger than the median due to more outliers
3. The median time is around 250-275, and the mean is more than that since the skew is on the right.

# Suppose many runners aim to finish in under four hours (240 minutes). Where would you look for evidence of this? How would this pattern show up in a histogram?

---

80 anonymous responses

**More people under 240 minutes**

14

Expresses the idea that there would be a greater number of runners with times under 240 minutes, phrased as a general increase.

**Many times just under 240**

12

Says there would be a concentration of finish times just below 240 minutes without explicitly using the word spike.

**Dip just before four hours**

11

Suggests there would be a noticeable decrease or trough in counts immediately before 240 minutes.

**Peak around four hours**

11

Claims there would be a prominent peak centered near 240 minutes, indicating many finish times clustering around that value.

# Response themes for question 2

---

**Histogram peak leans toward 240****10**

Indicates the overall mode or peak of the histogram would shift closer to 240 minutes than to smaller times.

---

**Look at proportion under 240****8**

Suggests assessing the fraction or proportion of runners finishing in less than 240 minutes as evidence.

---

**Spike just below four hours****8**

States there would be a noticeable peak or spike in the histogram immediately under 240 minutes, indicating many runners finishing just under four hours.

---

**Highest count at 240 minutes****6**

Asserts that the histogram bar corresponding to 240 minutes would have the largest count compared to other bars.

---

# Student responses

---

1. Because most runner finish around 200-300
2. expect more result under 240 minutes, and check the people who had  $\leq 240$  minutes
3. I would try to see the distribution of time they expect and the highest column should be less than 240
4. If people wanted to finish in under four hours, we would expect the frequency of results under 240 minutes to increase.
5. Most of finishing times should be less than 240 mins
6. Most of the bars would be before 240 minutes. People will start sprinting near the very end and their speed will go up.
7. The shape would change as more data would be below 240 minutes on the histogram. People will just slightly give it an extra push to go under 4 hours.
8. there could be more people under 240, shape would move towards around 240
9. There would be a higher frequency of runners showing up under the 240 minute mark.

# Student responses

---

1. We might expect to see the frequency in the 200-250 bucket increase a little bit and the frequency in the 250-300 range decrease a bit.
2. We would expect to see a disproportionate amount of runners finishing under the 240 minute mark. This would show up in a histogram with a right skew after 240.
3. We would expect to see alot of data around 240 minutes. It may not all be in that range, but I would expect many values to be at or below that range so as to beat that time
4. Would expect much of data to crowd around 240 minutes
5. You would expect more observations under 240 minutes and then a lot of observations above it.

# Student responses

---

1. I would for evidence of this if there's a high amount of people who finish between 235-239 minutes.
2. If there was a target of finishing before 240 minutes, there should be a larger amount of people finishing in the minute before that mark than in the minute after that mark
3. in the histogram we can see that there are lots in the range of 200-300
4. it is very hard, because it is far from mean
5. Many people would dip just before 4 hours.
6. That is true, most of the bars of frequency is before 400
7. the number of points right after 4 hours would be a lot less than the points right under 4 hours. (4:00-4:01 would be unproportionally smaller than 3:59-4:00)
8. There may be a lot of times just under four hours.
9. There might be a slight peak at 240 minutes or values around that number, we would see a dip after 240 minutes

# Student responses

---

1. there would be more data points just under 4 hours than just over 4 hours
2. You would expect more runners to finish in just under 240 minutes than just over it. There may be more people on a bar of people with 230 minutes range than 240.
3. you would look for evidence of this around the 230 to 260 minutes range. if theres a higher number around 240 you should see a spike there



# Student responses

---

1. A big dip right before 240
2. a lagre dropoff of data at a certian time and then a large spike at that point
3. Data under 4
4. I would find the time period of that, and count the time period
5. I would look for a skew in the data that might show that most people finish before 240 minutes.
6. If many runners aim to finish in under four hours, I'd look for runners who completed the race somewhere between four hours and three and a half hours. I'd look within a range, and maybe include a little above as well for those who couldn't finish in time.
7. If people wanted to finish under 4 hours, and they trained to get it, I think the data would skew to the left. The group with 200-250 would also probably increase in frequency.
8. the bars
9. The evidence can be found in the histogram as we can see that most runners are finished between 200 to 300 minutes. Therefore, they could have been practicing to finish in four hours

# Student responses

---

1. The left bin of the graph shows the people who finish before 240 mins.
2. there would be more people finishing in the 200-250 minute range with the size of the bins in the histogram.

# Student responses

---

1. I feel like a majority of those runners would be clustered around the four hour mark as some are able to make it and some aren't able to make it. I would look just under 240 minutes though/
2. I think there would a be a large frequency of runners in the 200-250 minute bucket.
3. I would expect to see a peak at 200 to 250 minutes in the histogram, and I would see clusters near the 240 value because the people who are running the marathon and providing me with the data are striving to hit the 240 minute benchmark.
4. I would look for a cluster at 240, and a dip on the higher side
5. I would see data concentrated a lot more around the 240 minute mark and then a steep dropoff in frequency following it
6. If people really wanted to finish under four hours, you would expect there to be a large jump in frequency right before the four hour mark, and fewer finishes right after 4 hours.
7. then most data should stack around 4hours.
8. There would be a peak around 240 minutes, there also may be a sharp drop in finish times after 4 hours
9. This pattern would show up in a histgoram more central to 240 minutes

# Student responses

---

1. We would expect to see most people right around that 240 minute mark with many being just before it because they will exert there selves to meet that 4 hour goal if it is possible
2. We would see a high frequency right around that 4-hour mark, and less frequency right after 4 hours because people will probably push themselves to get under the 4 hour mark.

# Student responses

---

1. I would look to see where the bulk finish. If the frequency is higher for times before 4 hours, there is evidence.
2. The average speed of runners later in the race who finish a little above, near or under the four hour time
3. The median would be closer to 240 and the distribution would be more normal.
4. The peak of the histogram will lean toward 240 more than before
5. This pattern would show as a right skew
6. volume around 4 hours
7. We should expect to see a large portion of people finishing just under 240 minutes because of the status they would get from this. We would look to see relative to other times if there is a peak there in the chart
8. You would look around the 200-300 bin amount from the previous histogram. You would want to check to see the number of people before this range and including the range, this will show you how likely it is that someone actually finished in under 4 hours
9. you would need to have more specific times on the histogram.

# Student responses

---

1. You would see the mean cluster around 240 minutes. The right tail would disappear as more people try to finish in under 240 min.

# Student responses

---

1. I would look at the proportion of people finishing less than 240 minutes on the plot
2. I would set a threshold line at  $t=240$  min, then check the proportion of data lying on the left of the threshold.
3. I would take the finish times within 20 minutes of the 4hr mark and compare the number of people 20 minutes under the 4hr mark to the number of people 20 minutes after the 4hr mark. If there is a large difference in these two then there might be a difference.
4. see how many finish before 240min
5. significant difference in the proportion of people from 3:50-4:00 and 4:00-4:10
6. You could create more bins so that you could see exactly the number of people who finish prior to 240 minutes, if a huge chunk of the people finish between 230-240 minutes you could say that they probably sprint at the end to beat that 240 minute time.
7. You could find evidence by looking at the proportions and see the concentration around the 4 hour mark

# Student responses

---

1. you would calculate how many people finished in under 240 minutes in proportion to the total amount of people who raced. this would show the majority of people finsihign in under 4 hours.



# Student responses

---

1. A spike in the histograms right under 240 minutes
2. I would look for a spike in the histogram just below 240 minutes, showing that many runners finished right before the four-hour mark. There might also be a drop in the number of runners finishing just after 240 minutes.
3. There should be a peak of data right before the 240 minute mark and then a dip right after.
4. There should be a spike just before 240 minutes and fewer people that finish just after 240 minutes
5. There would be a spike in just under 240 min
6. There would be a spike of individuals who finished just under 240 minutes, with dips before and after the time.
7. We would expect to see a spike for just under 4 hours. We can look for evidence of this by checking the histogram and expanding the number of bins, then looking for a spike in values right before 240.
8. you could look at the data, would show with a peak in the 240 minutes area

# Student responses

---

1. 240 minutes should have the highest count in the histogram. And almost all data should be under 240 minutes. Spike before 240 and dip right after.
2. I would look at the distrubtion pattern. Under 4 hours, 240 minutes would provide a histogram that is mostly right skewed
3. there would be a peak at 4 hours
4. We would expect to see the time taken per marathon decrease over the attempts of a given person until it leveled off below 4 hours. We'd want to run a spline and see if it's valid.
5. You can look at splits maybe, compare the goal times and actual fished times. it will show a spike right before the 250
6. You might see a high bar on a histogram for four hours because if that is everyones goal, then more people are going to get that time compared to if everyone just ran and had no goal. so maybe 240 would be close to the median/mean

- Filter the data to finishing times that are within ten minutes of four hours.
  - Plot a histogram of `finish_minutes``.
  - Label the axes appropriately.
- 

78 anonymous responses

Other responses

78

# Student responses

---

1. `finish_minutes <- marathon |> filter(finish_minutes > 230 & finish_minutes < 250) hist(finish_minutes, xlab = "Finishing Minutes", ylab = "Frequency")`
2. `library(dplyr) library(ggplot2) marathon %>% filter(between(finish_minutes, 230, 250)) %>% ggplot(aes(x = finish_minutes)) + geom_histogram(binwidth = 1, boundary = 240, fill = "steelblue", color = "white") + geom_vline(xintercept = 240, color = "red", linetype = "dashed") + labs(x = "Finish time (minutes)", y = "Number of runners", title = "Marathon finish times near four hours" ) + theme_minimal()`
3. `marathon`
4. `marathon filter(abs(finish_minutes - 60 * 4) < 10) ggplot() + geom_histogram(aes(x = finish_minutes))`
5. `marathon filter(abs(finish_minutes - 60 * 4) < 10) |> ggplot() + geom_histogram(aes(x = finish_minutes)) labs(title = "Dstribution of finish times", x = "Time", y = "count")`

# Student responses

---

1. `marathon %>% filter(finish_minutes >= 230, finish_minutes <= 250) %>% ggplot(aes(x = finish_minutes)) + geom_histogram(binwidth = 1) + labs(x = "Finishing time (minutes)", y = "Number of runners", title = "Marathon finishing times within 10 minutes of 4 hours" ) + geom_vline(xintercept = 240)`
2. `marathon |> filter(abs(finish_minutes - 240) <= 10) |> ggplot() + geom_histogram(aes(x = finish_minutes)) + geom_vline(xintercept = 240) + labs(x = "Finishing Time (minutes)", y = "Count")`
3. `marathon |> filter(abs(finish_minutes - 60 * 4) < 10) |> ggplot() + geom_histogram(aes(x = finish_minutes)) + labs(title = "Distribution of finish times", x = "Time", y = "count")`
4. `marathon |> filter(abs(finish_minutes - 60 * 4) < 10) |> ggplot() + geom_histogram(aes(x = finish_minutes))+ labs(title = "dist of times")`
5. `marathon |> filter(abs(finish_minutes - 60 * 4) < 10) ggplot() + geom_histogram(aes(x = finish_minutes))`
6. `marathon |> filter(abs(finish_minutes - 60 * 4) < 10) ggplot() + geom_histogram(aes(x = finish_minutes))`

# Student responses

---

1. `marathon |> filter(abs(finish_minutes - 60 * 4) < 10) |> ggplot() + geom_histogram(aes(x = finish_minutes)) + labs(title = "Distribution of finish times", x = "Time", y = "count")`
2. `marathon |> filter(abs(finish_minutes - 60 * 4) < 10) |> ggplot() + geom_histogram(aes(x = finish_minutes)) + labs(title = "Distribution of finish times", x = "Time", y= "count")`
3. `marathon |> filter(abs(finish_minutes - 60 * 4) < 10) |> ggplot() + geom_histogram(aes(x = finish_minutes)) + labs(title = "Distribution of finish times", x = "Time", y = "Count")`
4. `marathon |> filter(abs(finish_minutes - 60 * 4) < 10) |> ggplot() + geom_histogram(aes(x = finish_minutes))`
5. `marathon |> filter(abs(finish_minutes - 60 * 4) < 10) |> ggplot() + geom_histogram(aes(x = finish_minutes))`
6. `marathon |> filter(abs(finish_minutes - 60 * 4) < 10) |> ggplot() + geom_histogram(aes(x = finish_minutes)) labs( title = "Distribution of finish times", x = "Time", y = "Count" )`

# Student responses

---

1. `marathon |> filter(abs(finish_minutes - 60 * 4) < 10) |> ggplot() + geom_histogram(aes(x = finish_minutes)) + labs(title = "Distribution of finish times", x = "time", y = "count")`
2. `marathon |> filter(abs(finish_minutes - 60 * 4) < 10) |> ggplot() + geom_histogram(aes(x = finish_minutes)) + labs(title = "Distribution of finish times", x = "Time", y = "count")`
3. `marathon |> filter(abs(finish_minutes - 60 * 4) < 10) |> ggplot(aes(x = finish_minutes)) + geom_histogram() + geom_vline(xintercept = 240) + labs(x = "Finishing time (minutes)", y = "Number of runners")`
4. `marathon |> filter(abs(finish_minutes - 60 * 4) < 10) |> ggplot(aes(x = finish_minutes)) + geom_histogram() + labs(x = "Finishing time (minutes)", y = "Number of runners", title = "Marathon times within 10 minutes of 4 hours")`
5. `marathon |> filter(abs(finish_minutes - 60 * 4) < 10) |> ggplot() + geom_histogram(aes(x = finish_minutes)) + labs(title = "Distribution of finish times", x = "Time", y = "count")`

# Student responses

---

1. `marathon |> filter(abs(finish_minutes-60*4) < 10) |> ggplot() + geom_histogram(aes(x = finish_minutes))`
2. `marathon |> filter(abs(finish_minutes-60*4)<10)|> ggplot()+geom_histogram(aes(x=finish_minutes))|>`
3. `marathon |> filter(abs(finished_minutes - 60 * 4) < 10) |> ggplot() + geom_histogram(aes(x = finished_minutes)) labs(title = "Distribution of finish times around 4 hours")`
4. `marathon |> filter(finish_minutes >= 230 & finish_minutes <= 250) hist(marathon$finish_minutes)`
5. `marathon |> filter(finish_minutes >= 230 & finish_minutes <= 250) |> ggplot(aes(x = finish_minutes)) + geom_histogram(binwidth = 1) + geom_vline(xintercept = 240) + labs( x = "Finish time (minutes)", y = "Number of runners" )`
6. `marathon |> filter(finish_minutes >= 230, finish_minutes <= 250) |> ggplot(aes(x = finish_minutes)) + geom_histogram(binwidth = 1, boundary = 240, closed = "left") + geom_vline(xintercept = 240, color = "red") + labs( title = "Marathon finishes within ten minutes of four hours", x = "Finishing time (minutes)", y = "Number of finishers" )`



# Student responses

---

1. `marathon |> filter(finish_minutes >= 230, finish_minutes <= 250) |> ggplot(aes(x = finish_minutes)) + geom_histogram(binwidth = 1, fill = "lightblue", color = "white") + geom_vline(xintercept = 240, color = "red") + labs( title = "Marathon Finishing Times Near Four Hours", x = "Finishing Time (minutes)", y = "Number of Runners" )`
2. `marathon |> filter(finish_minutes >= 230) |> filter(finish_minutes <= 250) |> ggplot(aes(x = finish_minutes)) + geom_histogram(fill = "blue", color = "yellow", bins = 20) + geom_vline(xintercept = 240) + labs( x = "Finishing time (minutes)", y = "Number of runners", title = "Marathon finishing times near four hours" )`
3. `marathon |> filter(abs(finish_minutes - 60 * 4) < 10) |> ggplot() + geom_histogram(aes(x = finish_minutes)) + labs(title = "Distribution of finish times", x = "Time", y = "count")`
4. `marathon |> filter(abs(finish_minutes - 60 * 4) < 10) |> ggplot() + geom_histogram(aes(x = finish_minutes)) + labs(title = "Distribution of finish times", x = "time", y = "count")`

# Student responses

---

1. `marathon |> filter(abs(finish_minutes - 60 * 4) < 10) |> ggplot() + geom_histogram(aes(x= finish_minutes)) + labs(title = "Distribution of Finish Times", x = "Time", y = "Count")`
2. `marathon |> filter(abs(finish_minutrs - 60*4) < 10) |> ggplot() + geom_histogram(aes(x=finish_minutes))`
3. `marathon |> filter(abs(finish_minutes - 240) < 10) |> ggplot() + geom_histogram(aes(x = finish_minutes)) + labs(title = "Distribution of finish times", x = "Time (minutes)", y = "count")`
4. `marathon |> filter(abs(finish_minutes - 240) < 10) |> ggplot() + geom_histogram(aes(x = finish_minutes)) + labs(title = "Distribution of finish times")`
5. `marathon |> filter(abs(finish_minutes - 240) < 10) |> ggplot() + geom_histogram(aes(x=finish_minutes)) + labs(title="Distribution of finish time", x = "finish minutes around 4 hours", y="counts")`
6. `marathon |> filter(abs(finish_minutes - 60 * 4) < 10) |> ggplot() + geom_histogram(aes(x = finish_minutes))`
7. `marathon |> filter(abs(finish_minutes - 60 * 4) < 10) |> ggplot() + geom_histogram(aes(x = finish_minutes))`

# Student responses

---

1. `marathon |> filter(abs(finish_minutes - 60 * 4) < 10) |> ggplot() + geom_histogram(aes(x = finish_minutes)) labs(title = "Distribution of finish times", x = "Time" y = "Frequency")`
2. `marathon |> filter(abs(finish_minutes - 60 * 4) < 10) |> ggplot() + geom_histogram(aes(x = finish_minutes), bins = 10) + labs(xlab = Minutes to Finish a marathon, ylab = number of runners)`
3. `marathon |> filter(abs(finish_minutes - 60 * 4) < 10) |> ggplot() + geom_histogram(aes(x = finish_minutes))`
4. `marathon |> filter(abs(finish_minutes - 60 * 4) < 10) |> ggplot() + geom_histogram(aes(x = finish_minutes))`
5. `marathon |> filter(abs(finish_minutes - 60 * 4) < 10) |> ggplot() + geom_histogram(aes(x = finish_minutes)) + labs(title = "Distribution of finish times", x= "Time", y= "count")`
6. `marathon |> filter(abs(finish_minutes - 60 * 4) < 10) |> ggplot() + geom_histogram(aes(x = finish_minutes)) labs( x = "Finishing time in minutes", y = "Finishers" )`

# Student responses

---

1. `marathon |> filter(abs(finish_minutes - 60 * 4) < 10) |> ggplot() + geom_histogram(aes(x = finish_minutes)) + labs(title = "Distribution of Finish Times", x = "Time", y = "Count")`
2. `marathon |> filter(abs(finish_minutes - 60 * 4) < 10) |> ggplot() + geom_histogram(aes(x = finish_minutes)) + labs(title = "Distribution of Finish Times", x = "Time", y = "Count")`
3. `marathon |> filter(abs(finish_minutes - 60 * 4) < 10)|> ggplot() + geom_histogram(aes(x = finish_minutes))`
4. `marathon |> filter(abs(finish_minutes - 60 * 4) < 10) |> ggplot() + geom_histogram(aes(x = finish_minutes)) + xlab('time in minutes') + ylab('number of runners')`
5. `marathon |> filter(abs(finish_minutes - 60 * 4) < 10) |> ggplot() + geom_histogram(aes(x = finish_minutes)) |> labs(title = "Distribution of finish times", x = "Time", y="count")`
6. `marathon |> filter(abs(finish_minutes - 60 * 4)<10) |> ggplot() + geom_histogram(aes(x = finish_minutes))+ labs( x = "Finishing time (minutes)", y = "Number of runners" )`

# Student responses

---

1. `marathon |> filter(abs(finish_minutes - 60 * 4) < 10) |> ggplot() + geom_histogram(aes(x = finish_minutes)) + labs(title = "Distribution of finish times", x = "Time", y = "count")`
2. `marathon |> filter(abs(finish_minutes - 60 * 4) < 10) |> ggplot() + geom_histogram(aes(x = finish_minutes))`
3. `marathon |> filter(finish_minutes > 230, finish_minutes < 250) |> ggplot(aes(x = finish_minutes)) + geom_histogram() + geom_vline(xintercept = 240) + labs(x = "Finishing time (minutes)", y = "Number of finishes")`
4. `marathon |> filter(finish_minutes >= 230 & finish_minutes <= 250) |> ggplot() + geom_histogram(aes(x = finish_minutes)) + labs(x = "Minutes", y = "Number of Participants")`
5. `marathon |> filter(finish_minutes >= 230, finish_minutes <= 250) hist(marathon$finish_minutes, xlab = "Finish Time (minutes)", ylab = "Frequency")`
6. `marathon |> filter(finish_minutes >= 230, finish_minutes <= 250) |> ggplot() + geom_histogram(aes(x = finish_minutes)) + labs(x = "finish time (min)", y = "# of runners") + geom_vline(xintercept = 240)`

# Student responses

---

1. `marathon |> filter(abs(finish_minutes - 60 * 4) < 10) |> ggplot() + geom_histogram(aes(x = finish_minutes))`
2. `marathon |> filter(abs(finish_minutes - 240) < 10) |> ggplot(aes(x = finish_minutes)) + geom_histogram()`
3. `marathon |> filter(abs(finish_minutes - 60 * 4) < 10) |> ggplot() + geom_histogram(aes(x=finish_minutes)) + labs(title = "dist of finish times", x = "time", y = "count")`
4. `marathon |> filter(abs(finish_minutes - 60 * 4) < 10) |> ggplot() + geom_histogram(aes(x= finish_minutes)) + labs(title = "Distribution of finish times", x = "Time", y = "count")`
5. `marathon |> filter(abs(finish_minutes - 60 * 4) < 10) |> ggplot() + geom_histogram(aes(x = finish_minutes)) + labs(title = "Distribution", x = "Time", y = "count")`
6. `marathon |> filter(abs(finish_minutes - 60 * 4) < 10)) |> ggplot() + geom_histogram(aes(x = finish_minutes))`
7. `marathon |> filter(finish_minutes >= 230 & finish_minutes <= 250) |> ggplot() + geom_histogram(aes(x = finish_minutes)) + labs(title = "Distribution of Finish Times", x = "Time", y = "Count")`

# Student responses

---

1. `marathon |> filter(finish_minutes >= 230, finish_minutes <= 250) |> ggplot() + geom_histogram(aes(x= finish_minutes)) |> labs(title = "Distribution of finish times", x = "Time", y = "count")`
2. `marathon |> filter(abs(finish_minutes - 240) < 10) |> ggplot() + geom_histogram(aes(x = finish_minutes))`
3. `marathon |> filter(abs(finish_minutes - 60 * 4) < 10) ggplot() + geom_histogram(aes(x = finish_minutes)) |> labs(title = "Distribution of finish times", x = "Time", y = "count") |> geom_vline(xintercept = 240)`
4. `marathon |> filter(abs(finish_minutes-60*40)<10) |> ggplot() + geom_histogram(aes(x=finish_minutes))`
5. `marathon_240 <- marathon |> filter(finish_minutes >= 230 & finish_minutes <= 250) hist( marathon_240$finish_minutes, main = "Distribution of marathon finishing times", xlab = "Finishing time (minutes)", ylab = "Number of runners" ) geom_vline(xintercept = 240)`
6. `marathon|> filter(abs(finish_minutes-60*4)<10)|> ggplot() + geom_histogram(aes(x=finish_minutes))`

# Student responses

---

1. `marathon|> filter(abs(finish_minutes - 240) < 10)|> ggplot()+geom_histogram(aes(x=finish_minutes))> labs(title = "Distribution of finish times", x="time", y="Count")`
2. `marathon|> filter(abs(finish_minutes-240)<10)|> ggplot()+geom_histogram(aes(x=finish_minutes))+ labs( title = "Distribution of Marathon Finish Times Near 4 Hours", x = "Finish Time (minutes)", y = "Number of Runners" )`
3. `marathon|> filter(finish_minutes>=230, finish_minutes<=250)`
4. `marathon|>filter(abs(finish_minutes - 60*4)<10)|> ggplot()+geom_histogram(aes(x = finish_minutes)) help(geom_histogram)`
5. `near_four <- marathon %>% filter(finish_minutes >= 100, finish_minutes <= 300) ggplot(near_four, aes(x = finish_minutes)) + geom_histogram(binwidth = .05) + labs( x = "Finishing time (minutes)", y = "Number of runners" ) + geom_vline(xintercept = 240)`



# Student responses

---

1. `near_four <- marathon %>% filter(finish_minutes >= 230, finish_minutes <= 250) ggplot(near_four, aes(x = finish_minutes)) + geom_histogram(binwidth = 1) + labs( x = "Finishing time (minutes)", y = "Number of runners" ) + geom_vline(xintercept = 240)`
2. `timed <- marathon |> filter(minutes >= 230 & minutes <= 250) ggplot(data = timed, aes(x = minutes)) + geom_histogram()`
3. `withinten <- marathon |> filter(abs(finish_minutes - 60 * 4) < 10) |> ggplot() + geom_histogram(aes(x = finish_minutes))`

**Formulate a hypothesis about why there are more finishes just under four hours than just over four hours.**

---

0 anonymous responses

No responses were submitted.

- Add column ``minute`` by rounding ``finish_minutes`` down with ``floor()``.
  - Count the finishes in each ``minute``.
  - Arrange the rows in increasing order of ``minute``.
  - Create ``change = n - lag(n)``: this minute's count minus the previous minute's count.
  - Make a plot of ``change`` against ``minute``, with informative axis labels.
- 

79 anonymous responses

Other responses

79

# Student responses

1. `library(dplyr) library(ggplot2) marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> tidyr::complete(minute = 150:390, fill = list(n = 0)) |> arrange(minute) |> mutate(change = n - lag(n)) |> ggplot(aes(x = minute, y = change)) + geom_hline(yintercept = 0, color = "gray") + geom_line(na.rm = TRUE) + labs(x = "Finish time (minutes)", y = "Change in finishers from previous minute") + theme_minimal()`
2. `m <- marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate( minute = floor(finish_minutes) ) |> count(minute) |> arrange(minute) |> mutate( change = n - lag(n) ) plot(m$minute, m$change)`
3. `marathon %>% mutate(minute = floor(finish_minutes)) %>% count(minute) %>% arrange(minute) %>% mutate(change = n - lag(n)) %>% ggplot(aes(x = minute, y = change)) + geom_point() + geom_line() + labs( x = "Minute of finish time", y = "Change in number of finishes from previous minute" )`

# Student responses

---

1. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) library(dplyr) library(ggplot2) finish_changes <- marathon %>% mutate(minute = floor(finish_minutes)) %>% count(minute) %>% arrange(minute) %>% mutate(change = n - lag(n)) ggplot(finish_changes, aes(x = minute, y = change)) + geom_point() + geom_line() + labs(x = "Minute of finish", y = "Change in number of finishes from previous minute")`
2. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) mutate(minute = floor(finish_minutes)) |> count(minute, name = "n") |> arrange(minute) |> mutate(change = n - lag(n)) |> ggplot(aes(x = minute, y = change)) + geom_line() + labs(x = "Finish time (minutes)", y = "Change in number of finishes")`
3. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange(minute)`

# Student responses

---

1. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange(minute) |> mutate(ln = lag(n), change = n - ln) ggplot() + geom_line(aes(x = minute, y = change))`
2. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> mutate(ln = lag(n), change = n - ln)`
3. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange(minute) |> mutate(change = n - lag(n)) hist(marathon$minute)`
4. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) count(minute) |> mutate(lminute = lag(n), change = n - ln)`
5. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange(minute) |> mutate(change = n - lag(n)) |> ggplot() + geom_line(aes(x = minute, y = change))`

# Student responses

---

1. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange(minute) |> mutate(change = n - lag(n)) |> ggplot(aes(x = minute, y = change)) + geom_point() + geom_line() + labs( title = "Change in Marathon Finishers by Minute", x = "Finishing Time (minutes)", y = "Change in Number of Finishers" )`
2. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange(minute) |> mutate(lag_min = lag(n), dif = n - lag_min) |> ggplot() + geom_line(aes(x = minute, y = dif))`
3. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange(minute) |> mutate(ln = lag(n), change = n - ln) ggplot() + geom_line(aes(x = minute, y = change))`

# Student responses

---

1. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange(minute) |> mutate(ln = lag(n), change = n - ln) |> ggplot() + geom_line(aes(x = minute, y= change))`
2. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange(minute) |> mutate(ln = lag(n), change = n - ln) |> ggplot() + geom_line(aes(x = minute, y = change))`
3. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange(minute) |> mutate(ln = lag(n), change = n - ln) |> ggplot() + geom_line(aes(x = minute, y = change))`
4. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange(minute) |> mutate(ln = lag(n), change = n - ln) |> ggplot() + geom_line(aes(x = minute, y = change))`



# Student responses

---

1. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange(minute) |> mutate(ln = lag(n), change = n - ln) |> ggplot() + geom_line(aes(x = minute, y = change))`
2. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange(minute) |> mutate(ln = lag(n), change = n - ln) |> ggplot() + geom_line(aes(x = minute, y = change))`
3. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange(minute) |> mutate(ln = lag(n), change = n - ln) |> ggplot() + geom_line(aes(x= minute, y = change))`
4. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange(minute) |> mutate(ln = lag(n), change = n - ln) |> ggplot() + geom_line(aes(x=minute, y=change))`

# Student responses

---

1. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> mutate(ln = lag(n), change = n - ln)`
2. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> mutate(ln = lag(n), change = n - ln) |> ggplot() + geom_line(aes(x = minute, y = change))`
3. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> mutate(ln = lag(n), change = n - ln) |> ggplot() + geom_line(aes(x = minute, y = change)) lag(c(1, 2, 5, 10)) floor(2.5)`
4. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> mutate(ln = lag(n), change = n - ln) |> ggplot() + geom_line(aes(x = minute, y = change))`

# Student responses

---

1. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange(minute) |> mutate(ln = lag(n), change = n - ln) |> ggplot() + geom_line(aes(x = minute, y = change))`
2. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange(minute) |> mutate(ln = lag(n), change = n - ln) |> ggplot() + geom_line(aes(x = minute, y = change))`
3. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> mutate(ln = lag(n), change = n - ln) |> ggplot() + geom_line(aes(x = minute, y = change))`
4. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange(minute) |> mutate(ln = lag(n), change = n - ln) |> ggplot() + geom_line(aes(x = minute, y = change))`

# Student responses

---

1. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> mutate(change = n - lag(n)) |> ggplot() + geom_line(aes(x = minute, y = change))`
2. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> group_by(minute) |> summarize(n = n()) |> arrange(minute) |> mutate(change = n - lag(n)) |> ggplot(aes(x = minute, y = change)) + geom_point() + geom_line() + labs(x = "Minute of finish", y = "Change in counts from one minute to next")`
3. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange(minute) |> mutate(lminute = lag(n), change = n - lminute) |> ggplot() + geom_line(aes(x=minute,y=change))`

# Student responses

---

1. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate("minute" = floor(finish_minutes)) |> arrange(minute) |> count(minute) |> mutate(change = n - lag(n)) |> ggplot(aes(x = minute, y = change)) + geom_line() + labs(x = "Minute of finish time", y = "Change", title = "Minute-to-minute change in marathon finishes")`
2. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes))`
3. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute, name = "n") |> arrange(minute) |> mutate(change = n - lag(n)) |> ggplot(aes(x = minute, y = change)) + geom_point() + labs(x = "Finish Time (minutes)", y = "Change in Number of Finishes")`
4. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> mutate(ln = lag(n), change = n - ln) |> ggplot() + geom_line(aes(x = minute, y = change))`

# Student responses

---

1. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange(minute) |> mutate(ln = lag(n), change = n - ln)`
2. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange(minute) |> mutate(ln = lag(n), change = n - ln) |> ggplot() + geom_line(aes(x = minute, y = change))`
3. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange(minute) |> mutate(ln = lag(n), change = n - ln) |> ggplot() + geom_line(aes(x = minute, y = change))`
4. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange(minute) |> mutate(ln = lag(n), change = n - ln) |> ggplot() + geom_line(aes(x=minute, y=change))`

# Student responses

---

1. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange(minute) |> mutate(ln = lag(n), change = (n - ln)) ggplot() + geom_line(aes(x = minute, y = change))`
2. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange(minute) |> mutate(ln = lag(n), change = n - ln)`
3. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange(minute) |> mutate(ln = lag(n), change = n - ln) ggplot() + geom_line(aes(x=minute, y=change))`
4. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange(minute) |> mutate(ln = lag(n), change = n - ln) |> ggplot() + geom_line(aes(x = minute, y = change))`

# Student responses

---

1. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange(minute) |> mutate(ln = lag(n), change = n - ln) |> ggplot() + geom_line(aes(x = minute, y = change))`
2. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange(minute) |> mutate(ln = lag(n), change = n - ln) |> ggplot() + geom_line(aes(x=minute, y=change))`
3. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange(minute) |> mutate(ln = lag(n), change = n - ln) |> ggplot() + geom_line(aes(x = minute, y = change)) #slice_sample(n=10)`
4. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange(minute) |> mutate(ln = lag(n), change = n - ln) |> ggplot() + geom_line(aes(x = minute, y = change)) + xlab('finish time') + ylab('difference')`



# Student responses

---

1. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> mutate(ln = lag(n), change = n - ln) |> ggplot() + geom_line(aes(x = minute, y = change))`
2. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> mutate(ln = lag(n), change = n - ln) |> ggplot() + geom_line(aes(x=minute,y=change))`
3. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> mutate(ln = lag(n),change = n - ln) ggplot() + geom_line(aes(x = minute, y = change))`
4. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> mutate(ln =lag(n), change = n - ln)`
5. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange(minute) |> mutate(ln = lag(n), change = n - ln) |> ggplot() + geom_line(aes(x = minute, y = change))`

# Student responses

---

1. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange(minute) |> mutate(ln = lag(n), change = n - ln) |> ggplot() + geomline(aes(x = minute, y = change))`
2. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> mutate(lminute = lag(n), change = n - lminute)`
3. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange(minute) |> mutate(change = n - lag(n)) |> ggplot(aes(x=minute, y = change)) + geom_line() + labs(x = "finish minute" , y = "change in count from prev min")`
4. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange(minute)|> mutate(lminute = lag(n), change = n-lminute) |> ggplot() + geom_line(aes(x=minute, y=change))`

# Student responses

---

1. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange(minute) |> mutate(ln = lag(n), change = n - ln) ggplot() + geom_line(aes(x = minute, y = change))`
2. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391)|> mutate(minute = floor(finish_minutes))|> count(minute) |> arrange(minute) |> mutate(change = n-lag(n)) |> ggplot() + geom_line(aes(x=minute, y = change))`
3. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391)|> mutate(minute = floor(finish_minutes))|> count(minute)|> arrange(minute)|> mutate(ln = lag(n),change = n-ln)|> ggplot() + geom_line(aes(x=minute, y = change))`
4. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391)|> mutate(minute = floor(finish_minutes) )|> count(minute)|> mutate( ln = lag(n), change = n - ln)|> ggplot()+geom_line(aes(x = minute, y = change))`

# Student responses

---

1. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391)|> mutate(minute = floor(finish_minutes))|> count(minute)|> arrange(minute)|> mutate(ln = lag(n), change = n - ln)|> ggplot()+geom_line(aes(x = minute, y = change))`
2. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391)|> mutate(minute = floor(finish_minutes))|> count(minute)|> mutate(ln=lag(n), change = n-ln)|> ggplot()+geom_line(aes(x=minute, y=change))`
3. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391)|> mutate(minute = floor(finish_minutes))|> count(minute)|> arrange(minute)|> mutate(ln = lag(n), change = n - ln)|> ggplot(aes(x = minute, y = change)) + geom_line() + labs( x = "Minute of finish time", y = "Change in number of finishes from previous minute" )`
4. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391)|> mutate(minute=floor(finish_minutes))|> count(minute)|> arrange(minute)|> mutate(ln = lag(n), change = n - ln)|> ggplot()+geom_line(aes(x = minute, y = change))`

# Student responses

---

1. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391)|> mutate(minute=floor(finish_minutes))|> group_by(minute)|> summarize(n=n())|> arrange(minute)|> mutate(change = n - lag(n))|> ggplot()+geom_line(aes(x=minute,y=change))+ labs( title = "Distribution of Marathon Finish Times Near 4 Hours", x = "Finish Time (minutes)", y = "Change: this minute's count minus the previous minute's count" )`
2. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391)|> mutate(minutes = floor(finish_minutes)) |> count(minutes) |> arrange(minutes) |> mutate(ln = lag(n), change = n - ln) |> ggplot() + geom_line(aes(x=minutes, y=change))`
3. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391)|> mutate=(minute=floor(finish_miuntes))|> count(mutate)|> mutate(n=log(n))`

# Student responses

---

1. `marathon |> filter(finish_minutes >= 150, #exclude the outliers finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange(minute) |> mutate(ln = lag(n), change = n - ln) |> ggplot() + hist(aes(x = minute, y = change))`
2. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange(minute) |> mutate(change = n - lag(n)) |> ggplot(aes(x = minute, y = change)) + geom_col() + geom_vline(xintercept = 240, color = "red") + labs( title = "Change in finishes from the previous minute", x = "Minute", y = "Finishes this minute minus previous minute" )`
3. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange(minute) |> mutate(ln = lag(n), change = n - ln) |> ggplot(aes(x = minute, y = change)) + geom_point() + geom_line() + labs( x = "Minute of finish", y = "Change in number of finishes from previous minute" )`

# Student responses

---

1. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> mutate(ln = lag(n), change = n - ln)`
2. `marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> group_by(minute) |> summarize(n = n()) |> arrange(minute) |> mutate(change = n - lag(n)) |> ggplot(aes(x = minute, y = change)) + geom_point() + geom_line() + labs(x = "Minute of finish", y = "Change in number of finishes from previous minute")`
3. `marathon |> filter(finish_minutes >= 159, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange (minute)`

# Student responses

---

1. `marathon_changes <- marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange(minute) |> mutate(change = n - lag(n)) ggplot(marathon_changes, aes(x = minute, y = change)) + geom_point() + labs( x = "Minute of finish time", y = "Change in number of finishes from previous minute" )`
2. `minute_counts <- marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange(minute) |> mutate(change = n - lag(n)) plot(minute_counts$minute, minute_counts$change, xlab = "Finishing Minute", ylab = "Change in Finishes vs. Previous Minute", type = "l")`
3. `new <- marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> count(minute) |> arrange(minute) |> mutate(change = n - lag(n)) ggplot(new, aes(x = minute, y = change)) + geom_point() + labs( x = "Finish time (minutes)", y = "Change in runner count" )`



# Student responses

---

1. `outie <- marathon |> filter(finish_minutes >= 150, finish_minutes < 391) |> mutate(minute = floor(finish_minutes)) |> group_by(minute) |> summarize(num_mins = n()) |> mutate(change = num_mins - lag(num_mins)) ggplot(outie, aes(x = minute, y = change)) + geom_point() + labs( x = "Minute", y = "Change in number of finishes from previous minute" )`

# Summarize the findings of your data exploration in a few sentences, highlighting the patterns you observed and any potential explanations for these patterns. Is there any limitation to your analysis?

---

81 anonymous responses

## Clustering just before time goals

25

Reports that finish times cluster immediately before round-number or milestone times, indicating many aim to finish just under targets.

## Spikes at round or half-hour numbers

15

States that large spikes occur specifically at round-number times like whole or half hours, supporting the milestone hypothesis.

## Specific dip near 240 minutes

12

Notes a pronounced negative change or drop in finish counts around 240 minutes (four hours).

## Runners speed up approaching hour marks

11

Observes increased pace or sprinting behavior as runners near hour or round-minute milestones, causing spikes just before those marks.

# Response themes for question 6

---

**Notes missing labels or incomplete data**

6

Highlights that trends may be biased or unclear because some entries lack labels or necessary metadata.

---

**Unclear or empty response**

5

Responses that contain no substantive reasoning or are empty.

---

**Mentions outliers or data issues**

4

Points out limitations of the analysis such as outliers affecting results or other data quality concerns.

---

**Spikes linked to specific target hours (e.g., 5h)**

3

Identifies spikes tied to particular target times (like under five hours or 499 minutes) showing many aim for those cutoffs.

---

# Student responses

---

1. Based on the data and the "lag" spikes there is some evidence suggesting that more runners tend to finish before a big time milestone than after.
2. found spikes of data at certian values that are usual round number or numbers that end in zero and at 9 to be under a certian hour mark
3. I noticed that there are more marathon finishes right before round-hour marks, especially around four hours, followed by a drop right after. One possible explanation is that runners may have a goal of finishing under a certain round time, such as four hours, so they push themselves to finish just before that mark.
4. In the data set, we found that there are certain values where the change in frequency is greater, suggesting that there may be target times for which runners want to finish in
5. In the exploration of our data we saw that there is a tendency for people to finish right before very round numbers (2.5 hours, 3 hours...). We saw this in the data because there are large spikes in people finishing right before the time passes and it kept showing up and was not a fluke at one time.

# Student responses

---

1. It seems like the largest spikes in the plot were around the areas which were major times such as 4 hours, 3.5 hours, 4.5 hours. This could probably because people are looking to be able to say they went faster than a certain time.
2. Marathon finishing times show noticeable changes around round-number target times, especially four hours. There are more runners finishing just before four hours than just after it, and patterns may occur around other common finishing-time goals. However, this is observational data.
3. One key observation in the dataset is that the largest spike in change from how many runners finish in one minute to the next minute was the moment before the 4 hour mark, so runners speed up approaching certain marks like the end of an hour
4. overall the histogram has a lot of variability with few key things to point out. there was a big downward spikes, with the biggest one at around 240. This means alots of runners push hard to finsih just under a milestone, so the minute before it (3:59) is packed. the next minute (4:00-4:01) is way emptier.

# Student responses

---

1. People generally saw clear improvements in their marathon times as they got closer to 240.
2. people start sprinting if they are close to milestone times. for example right before 4 hrs/3 hrs/etc.
3. people want to finish around the 10s when running, particularly 240; limitation is they dont reach it
4. The data shows that more runners finish just before major time goals, such as four hours, than just after them. This may be because runners push themselves to reach these goals. However, we cannot confirm their motivations from finishing times alone.
5. The findings of our data explores how people may run races with the goal in mind of breaking common barriers, such as hours or half-hour marks. There is some limitation because of coincidence, or setting a floor which might make times farther apart than they really seem.
6. There were spikes in the amount of people finishing the marathon at whole times showing that people were more eager to finish the race before these times.

# Student responses

---

1. We found that before several key time milestones there are large spikes in the number of finishers. This is likely due to peoples' desire to achieve a certain time.
2. We found that finish times spiked around round number like 4 hours, 220 mins, etc. This happened because runners aim for certain running times and pace themselves accordingly
3. We found that people seem to be rushing just before to make it into certain time cut offs.
4. We found that there are certain arbitrary times, such as 210, 240, and 270 minutes that see abnormal spikes just under that time threshold, indicating people are attempting to finish just under those spikes.
5. We found that there is a spike at 240 minutes because that explains the behavior of racers trying to achieve that goal, and then there is a dip that follows because most people have completed the race by then. We are limited by the people that did not try to achieve 240 but happened to get 240

# Student responses

---

1. We found that there tends to be spikes in finishing times around certain minute marks, like 240 minutes and 299 minutes. This may be because people tend to push themselves to finish under a certain time.
2. We found that there was a lot of data within the range a round 240 minutes. There is limitation because there is no knowledge of the goals that people actually had and whether or not they were going for it.
3. We observed that many people finish just before 240 minutes (or 4 hours), which could likely be explained by the psychological element of wanting a time sub 4 hours. There is a potential limitation of seconds
4. We saw a huge spike in the graph (clustering) of where 240 minutes was, indicating how many people tried their best to reach that under 240 minutes threshold.
5. We saw spikes of individuals finishing just under "whole" times like hour thresholds.



# Student responses

---

1. A lot of people try to finish the marathon in under 5 hours, which can be seen by the increasing number and the spike in numbers at 499 minutes.
2. humans have a natural instinct to set goals for themselves and push themselves to reach those goals. we see this by spikes in finishes around certain numbers like 3.5 4 and 5 hours
3. I found that more rounded marathon times, like 4 or 5 hours, have a higher frequency because people are motivated to finish under certain cutoffs
4. Marathon finishing times are right-skewed, with most runners clustered around four to four and a half hours and a long tail of slower finishers. Limitations: this is observational data, so it shows the pattern but does not prove why any individual sped up.
5. Noticed that most people finish under 400 minute mark, could be due to final push to get a rounded finish time.
6. People tend to finish marathons at whole hours/ right before an hour mark. This makes sense because it is a common goal to get under a certain hour. One limitation is if these are official times because people may round.

# Student responses

---

1. People tend to run into 240 minutes and other whole hours
2. The data shows that runners prefer to finish at specific times linked to half or whole hours.
3. the patterns I observed is that around 250 the wave is much more larger than others
4. There appeared to be sudden jumps where people would finish all at once during a time goal
5. There is a large spike in finishers right around the 240 minute mark, which means many people are trying to finish around that time compared to others.
6. There were spikes in the data especially right before the 300 mark because people were trying to not hit that
7. We found large spikes at round numbers, likely due to our hypothesis. This test is limited; we cannot fairly attribute our hypothesis to these results without a test.
8. We found that people tend to run faster to get to a specific hour point for their marathon.

# Student responses

---

1. We found that there are specific times that runners aim to achieve during marathons. We went through differneces in times acorss every 1 minute difference between all the possible marathon times. In conclusion, we found several "spikes" showing that there were noticebale difference sbetween marathon times, showing various runners goals

# Student responses

---

1. Before the four hour mark there is a big change in time.
2. Change: this minute's count minus the previous minute's count, drops largely at minutes=240.
3. difference is bigger around 240
4. I found there was a very negative change near 240, some on other points. Probably near things like 210 and such where people care about getting sub-certain milestones
5. it is hard to exceed 250 mins
6. It was observed that there is a large difference in number of marathon runners that ran under a rounded time, for example under 240 minutes. This may be because of runners' goals.
7. The graph had a huge spike around 240 negatively, meaning that the runners rushed toward the finishing line once it reached 4 hours timemark
8. There are a few spikes, noticeably right around the 240 minute mark. Limitations in the analysis are the choice to make the time comparisons 1 minute, any time interval could be used.
9. There is a drastic amount of changes in finishes near 240 minutes, which suggests runners aim for that time.

# Student responses

---

1. These findings suggests that for some two minute intervals, there is a drastically large change of people who finish in one minute verses the other. A limitation could be that this interval doesn't look at three or four minute intervals which might also have a big difference.
2. We found a pattern where the frequency of finish times decreased significantly before major milestones, such as 2 hours. This could be because people who time their races in real time want to make under a certain threshold, and so anyone who would normally finish right before an hour marker would push themselves to be under it.
3. We found that there's high spike in four hours or 240 minutes. We found it through analyzing the people who finished around the 240 minutes and taking a closer look into it.

# Student responses

---

1. I found that there are spikes just before 240 and after and this must be that people are noticing that hey i am just about to make it to 4 hours so i will sprint so I can make it and that is why we see the discrepancy in the data
2. i found that there was a trend in runners increasing their speed before the finish line when approaching an hour mark
3. So a lot of people try to made in that time when they are near 240min goal. If they cant. They kinda give up.
4. The changes from the previous minute appear to be biggest around certain hour and hour and a half finish times. This could be because runners are shooting for a sub 4 hour or sub 3:30 marathon, and push to finish under those times.
5. the data showed large spikes right before hour and half hour time marks followed by large dips right after the marks. This is likely explained by runners speeding up when they are close to these marks to finish under the time as opposed to over.
6. There is a high level of change between some minutes vs others. This is likely do to some milestones being close so people push harder to get there.

# Student responses

---

1. There is a higher level of change between some minutes versus other, this is probably because there are time milestones that people sprint for before their race ends.
2. We found 'spikes' of finishers around hour marks (3 hrs, 4hrs, etc). It is possible that people are sprinting to the finish to make it in under \_\_ hours. However, correlation doesn't equal causation
3. We found that runners burn out right before the end to be below the time at the end
4. We observed that on minutes that are whole numbers, for example 2 hours (120), 3 hours (180), 3 hours 30 min (210), there is a downward spike before the minute showing more people finish before that mark
5. We observed that there was the largest change in marathon finishes at even numbers such as 240 minutes. The potential explanation is that runners have a goal set in mind (usually these nice even numbers) and speed up at the end to reach that goal

# Student responses

---

1. I feel like some people are trying to go for certain times whereas others are running it for fun. It is hard to tell peoples reasoning behind running the marathon
2. I indentify a decreasing trend. I think a clear limitation is that it fails to capture those without labels
3. I'm still confused on the lag operator. I think it gives you the amount of people ahead of you in the line of an ascending list.
4. in the data, there are certain even-numbered times where the change is greatest, implying that people race quickly to finish under these times. The limitation of this is that this correlation does not necessarily imply causation
5. We see patterns at key time periods where people are likely to push themselves to finish sooner. The limitation of our data is not much
6. Yes there was spike skn the grah and we saw in the data that there was indeed a push for a specifc time which gets distinguished and noticable when we look at graph. The limitation is that we dont actually know what the runner sgoal time is so we don't know what is going on mentally see if they try to mkae a final push at the end



# Student responses

---

1. .
2. no
3. not sure
4. We found that the change around 240 fluctuate the most. Limitation: add more interval
5. yes, others may not understand it

# Student responses

---

1. Changes always remain near zero whatever the time is. That means, the distribution is almost uniform over time. However, there are two limitations. Firstly, the second row of the table recording the first pair of data has NA in change. Secondly, the scale is too large.
2. I found that the limitations are outliers
3. the data suggest that people try to get their time under certain barriers
4. There seems to be big jumps around every 50-60 minutes. This might suggest that people are going to shoot for a number just below a time target that is a whole hour. The limitation to this analysis is a lack of other survey information.

# Student responses

---

1. dont think there are any potential explanations
2. Less and less people run lower
3. The data is relatively consistent over time, but there are certain spikes in either direction.

# What pattern do you expect to see if we plot the number of births for each day of the week? Why do you expect that pattern?

---

83 anonymous responses

## Approximately equal across weekdays

46

Students expect birth counts to be roughly the same for each day, assuming no strong day-specific effects or randomness leading to an even distribution.

## More weekday births due to scheduled procedures

11

Students expect higher counts on weekdays because hospitals schedule inductions and C-sections during business days, reducing weekend births.

## Slightly more weekend births

9

Students think scheduled procedures might lead to a small increase in weekend births, though they expect the effect to be minor.

## Other responses

8

Responses the model could not place reliably.

# Response themes for question 7

---

|                                                                                                         |   |
|---------------------------------------------------------------------------------------------------------|---|
| Other responses                                                                                         | 6 |
| Uneven or nonuniform distribution                                                                       | 3 |
| Students assert the distribution will not be uniform, without specifying a particular pattern or cause. |   |

# Student responses

---

1. 7 very roughly equal values.
2. Approximately equal across the weekdays. I expect this because this trend will occur because
3. evenly distributed, should be pretty random
4. Expect uniform distribution, because should be nothing that makes it not all evenly distributed across days
5. I do not expect to see a pattern as birth dates should be completely random
6. I don't expect to see a pattern for each day of the week, since births are pretty random and don't happen more often at a certain time or weekday
7. I don't expect to see much of a pattern, but it is possible that weekdays are more common since c-sections or inductions might be scheduled for those days
8. I don't think we would see a pattern for plotting the number of births for each day of the week, because it is random, and birth is also random, leading to too much variation for a pattern to be clear.
9. I don't think we would see any crazy pattern. It will probably look very uniform and even across each day of the week.

# Student responses

---

1. I dont expect to see a pattern
2. i expect a fairly even amount on each day, because the day of the week has nothing do do with when a mother goes into labor
3. I expect a uniform distribution as birth day of the week seems uniformly randomly distributed dependent as pregnancy start date does not entirely predict delivery.
4. I expect evenly distributed pattern. Since birth could happen every day in the week.
5. I expect that the number of births will be mostly uniform for each day of the week, as a baby is equally likely to come on any day of the week. There could perhaps be some spikes on the weekday, as people may be more likely to be induced then.
6. I expect the number to be uniform across the wek
7. I expect the plot to be relatively uniformly distributed as this is a natural process with little human control.
8. I expect there to be roughly the same amount of births on each day of the week because babies are mostly born at random within small time frames like a week

# Student responses

---

1. I expect to see a uniform distribution for each day of the week because you can't really choose when you give birth. Even with inducing labor, that's usually a day the doctor recommends.
2. I expect to see a uniform plot if we plotted the number of births for each day of the week. This is because I don't expect a correlation between the day of the week and an inclination to give birth to a child.
3. I expect to see no pattern honestly
4. I feel like the babies should be pretty equally distributed among the days of the week because you don't really get to choose when the baby comes, it just comes. I am curious if the weekend would be higher because people can now induce and I think a weekend induction makes sense, since people don't have to worry about work.
5. I think births will be evenly distributed across days of week
6. I think it should be uniform, considering the probability of generating a baby is the same



# Student responses

---

1. I think they should all be pretty equal, just because I can't think of any reason why a naturally occurring birth would happen on a certain day. for cesarean sections maybe more would be scheduled for weekdays, so maybe some more on weekdays
2. I think we would see a lot of cluttered data across 7 points since there are only 7 days a week.
3. I will keep supporting when plotting the numbers of birth for each week. I expect to see it very similar, mainly focus on holidays
4. I would expect a uniform distribution, and no patterns whatsoever because I can't think of any reasons of why there would be a pattern.
5. i would expect an equal distribution across each day. about  $\frac{1}{7}$  of babies should be born each day of the week. because days constantly changing which date they are each year
6. I would expect birthdays to be evenly distributed among every day.
7. I would expect each day of the week to have the same number of births because hypothetically each day has the same likelihood chance
8. I would expect it to uniformly distributed

# Student responses

---

1. I would expect the number of births for each day of the week to not have a distinct pattern because the birth of babies near the due date tends to fluctuate
2. I would expect the number of births for each day of the week to not have a pattern because people cannot control when they go into labor.
3. I would not expect to see any major pattern, since variations in the gestation period scatter the pattern.
4. It should be roughly equal for each day of the week. This is because there is no reason to birth on any other day of the week
5. It should be roughly uniform for all days of the week because there really isn't anything stopping that from happening.
6. it would be evenly distributed since it is random among weekdays less likely on weekends
7. My expectation if we plot the number of births per each day of the week would be a uniform distribution. People don't really plan to give birth, maybe if it was a C-section, but it should be evenly distributed across days.
8. Should be average
9. symmetric because there shouldn't be a difference

# Student responses

---

1. There should be little to no pattern for the number of births each day of the week because there are no predictable dates for when a baby would be born.
2. There should be uniform distribution of births but a pattern might emerge if you consider women who use an epidural or C-section and have more control of birth day.
3. uniform
4. uniform, but more on weekday
5. We may see a uniform distribution between 7 days.
6. within a given week, i would expect births to fairly evenly distributed

# Student responses

---

1. Expect to see more births for weekdays than weekends because surgeries tend to be scheduled during weekdays
2. I expect more during the weekdays and friday because there are more doctors available on the weekdays and they try to get babies out before the weekend
3. I expect that there will be more births on weekdays because of the times when hospitals are open. Any women who are going into induced labor or having a c-section should give birth on a weekday.
4. I expect there to be less births occurring on the weekends and the births throughout the week to be pretty consistent
5. I would expect to see the number of births for each day of the week to be about uniformly distributed possibly with a slight increase on weekdays as people try to avoid stressful situations/places on the weekends as well as hospitals will likely have more staff during those times
6. I would maybe expect to see slightly more births on weekdays, especially with planned c-sections.
7. I would see fewer births on weekends and more on weekdays. This may be because scheduled births

# Student responses

---

1. Maybe more babies being had during the weekdays, with perhaps some outside pressure from the hospital to have the child during times when more doctors are in
2. My expectation is fewer births on the weekend because there is less scheduling of hospitals on weekends
3. Surge from Mon-thur
4. the amount of births per week will peak from M-F. I'd assume this is because this is all of the sheduled C-section births that follow the tradiitional M-F workweek.

# Student responses

---

1. I do not expect that there will be a super large pattern early in the data set about days of the week and the number of births. I do think that later on in the data set we may see more births occurring on weekend type days because of people being able to schedule for things like a c-section or being induced and you get more selection
2. I don't think they would be any differences however i think the weekend may have more births as they are more likely to be unstressed and have actual time to have someone drive them to the hospital
3. I expect more babies to be born on the weekends since that's when people are free more often to have both parents be there.
4. I expect the weekends to have more for medically controlled births when it was not the due date affecting the data. I do expect it to be somewhat equal because medical births are not common.
5. i hypothesize more people are born on weekends than weekdays since that is more convenient to go to the hospital and give birth.
6. I would expect there to be a relatively even amount of births on every day with maybe slightly more people on the weekend

# Student responses

---

1. I would expect there to be very slightly more births on the weekends, as people who schedule c-sections may do it when they are not working. It should not be a significant impact however
2. Maybe weekends would be more counts than weekdays due to scheduled procedures
3. More births on weekends since people are free to go to the hospital

# Student responses

---

1. i can not find a pattern
2. I didn't;t expect the pattern
3. I expect to see a higher proportion of births on saturday and sunday because parents may plan c-sections when they are less busy on the weekends.
4. I expect to see more births on the weekends, because with medical technology people are able to induce births, and so they would prefer a weekend birth to avoid spouses missing work, etc.
5. I expect to see more births on weekdays
6. I expoect to see unirfomr distribution of births druing the week, we shoudl be looking at days of teh week, we should be looking at that as well as month of the year
7. it will have peak at Tue
8. We woudl expec tahtt to be similar acrosos days categorising



# Student responses

---

1. I expect more births on weekdays than on weekends because hospitals often schedule induced labor and C-sections during the week. Births may be less common on Saturdays and Sundays when fewer procedures are scheduled.
2. I think mainly weekends, but by a very slight amount.
3. I think that births being more likely on the weekdays would make sense because stress can be a factor when it comes to a person starting labor.
4. I think we will see more births around november time and this is because of februaray and you probably can guess why. That month has a lot of potential for babies to be conceived so I think we will see a large increase towards the end months after 9 month waiting for each day maybe the same
5. I would expect to see a mostly uniform distribution, with maybe a higher frequency on weekdays, since it is easier to see a doctor on those days.
6. should me flattern since you cannot decide when to birth

# Student responses

---

1. I expect an ununiform distribution with some specific days having a larger number than others. This is because the seven days of a week isn't evenly distributed within a year. Some days have 52 while others have 53.
2. If we plot the number of births for each day of the week, I think we won't see any pattern because I think it is more correlated with months or years than days
3. not uniformly distributed

- Group ``births`` by ``day_of_week``. - Calculate ``mean_births``, the average of ``births`` for each weekday. - Save the seven-row result as ``weekday_births``. - Use ``geom_col()`` to plot one bar per weekday, with height given by ``mean_births``. - Label the axes to show the weekday and average number of births per day.

---

80 anonymous responses

Other responses

80

# Student responses

---

1. `births |> group_by(day_of_week) |> summarise(mean_births = mean(births)) |> ggplot(aes(x = day_of_week, y = mean_births)) + geom_col( labs(x = "day of week", y = "Avg. Births")) )`
2. `births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> ggplot() + geom_col(aes(x = day_of_week, y = mean_births)) + labs( x = "Day of week", y = "Average births per day" )`
3. `births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x = day_of_week, y = mean_births)) + labs( x = "Day of week", y = "Average births per day" )`
4. `births |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> group_by(day_of_week, weekend) |> summarize(mean_births = mean(births)) |> ggplot() + geom_col(aes(x = day_of_week, y = mean_births, fill = weekend)) labs(x = "Day of week", y = "Avg_Births")`

# Student responses

---

1. `births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() +  
geom_col(aes(x = day_of_week, y = mean_births, fill = weekend))`
2. `births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() +  
geom_col(aes(x = day_of_week, y = mean_births, fill = weekend)) + labs(title = "Births Based On Day of Week", x = "Day Of Week", y = "Mean Births")`
3. `births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() +  
geom_col(aes(x = day_of_week, y = mean_births, fill = weekend)) + labs(title = "Average Number Of Births per Weekday", x = "Day of week", y = "Average  
number of births" )`
4. `births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() +  
geom_col(aes(x = day_of_week, y = mean_births, fill = weekend)) + labs(x = "week day", y = "average births")`

# Student responses

---

1. `births |> group_by(day_of_week)|> summarize(mean_births = mean(births))|> ggplot(aes(x = day_of_week, y = mean_births)) + geom_col() + labs( x = "Day of week", y = "Average number of births per day" )`
2. `births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> mutate(weekend = day_of_week %in% c("Sat", "Sun"))|> ggplot() + geom_col(aes(x = day_of_week, y = mean_births, fill = weekend)) + labs(x = "Day", y = "Average Births per Day")`
3. `births |> group_by(day_of_week) |> summarize(mean_births = mean(births))|> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x = day_of_week, y = mean_births, fill = weekend)) + labs(x = "Day", y = "Avg Births")`
4. `births |> group_by(day_of_week) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> summarize(mean_births = mean(births)) |> ggplot() + geom_col(aes(x = day_of_week, y = mean_births)) + labs(x= "Day of Week", y = "Average number of births")`

# Student responses

---

1. `births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> ggplot() + geom_col(aes(x = day_of_week, y = mean_births))`
2. `births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> ggplot() + geom_col(aes(x = day_of_week, y = mean_births)) + labs(x = "Weekday", y = "Mean births")`
3. `births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x = day_of_week, y = mean_births, fill = weekend))`
4. `births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x = day_of_week, y = mean_births, fill = weekend)) + labs(x = "day of week", y = "average # of births")`

# Student responses

---

1. `births |> group_by(day_of_week) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> summarize(mean_births = mean(births)) |> ggplot() +  
geom_col(aes(x = day_of_week, y = mean_births), fill = weekend) + labs(x = "day of week", y = "avg births")`
2. `births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) -> weekday_births  
weekday_births |> ggplot(aes(x=day_of_week, y = mean_births)) + geom_col() + labs(x= "Day of Week", y = "mean births per day of week")`
3. `births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(  
aes(x = day_of_week, y= mean_births, fill = weekend))`
4. `births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(  
aes(x=day_of_week, y=mean_births, fill = weekend)) + labs(x = "Day of week", y = "Avg. births")`



# Student responses

---

1. `births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() +  
 geom_col(aes(x = day_of_week, y = mean_births, fill = weekend)) + labs(title = "Birth per days of the week", x = "Day of Week", y = "Average Births")`
2. `births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(  
 aes(x=day_of_week, y =mean_births, fill = weekend)) + labs(x = "Day of Week", y = "Avg number of births")`
3. `births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() +  
 geom_col(aes(x = day_of_week, y = mean_births, fill = weekend)) + labs(x = "Day of Week", y = "Avg Births")`

# Student responses

---

1. `births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() +  
 geom_col(aes(x = day_of_week, y = mean_births, fill = weekend)) + labs(x = "Day of week", y = "Avg births")`
2. `births |> group_by(day_of_week)|> summarize(mean_births = mean(births))|> mutate(weekend = day_of_week %in% c("Sat", "Sun"))|> ggplot()+geom_col(  
 aes(x = day_of_week, y = mean_births, fill = weekend))+ labs(x = "Day of week", y = "Avg. births")`
3. `births |> #head() group_by(day_of_week) |> summarize(mean_births = mean(births)) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() +  
 geom_col( aes(x = day_of_week, y = mean_births, fill = weekend)) + labs(x = "Day of Week", y = "Average Number of Births" )`

# Student responses

---

1. `births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> mutate(weekend = day_of_week %in% c("Sat", "Sun"))|> ggplot() + geom_col(aes(x = day_of_week, y = mean_births, fill = weekend) ) labs(x = "Day of week", y = "Avg. births")`
2. `births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> ggplot(aes(x = day_of_week, y = mean_births)) + geom_col() + labs( x = "Weekday", y = "Average number of births per day" )`
3. `births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x = day_of_week, y = mean_births, fill = weekend)) + labs(x = "Day of week", y = "Avg. births")`
4. `births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x = day_of_week, y = mean_births, fill = weekend)) + labs(x = "Day of week", y = "Avg. births")`

# Student responses

---

1. `births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) \> print() ggplot() +  
geom_col(aes(x = day_of_week, y = mean_births, fill = weekend)) + labs(x = "Day of week", y = "Avg Births")`
2. `births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() +  
geom_col(aes(x = day_of_week, y = mean_births, fill = weekend)) + labs(x = "day of week", y = "Avg. births")`
3. `births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> mutate(weekend = day_of_week %in% c("Sat", "Sun"))|> ggplot() + geom_col(  
aes( x = day_of_week, y = mean_births, fill = weekend))`
4. `births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> mutate(weekend = day_of_week %in% c("Sat","Sun")) |> ggplot() + geom_col(  
aes(x = day_of_week, y = mean_births, fill = weekend))`

# Student responses

---

1. `births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> mutate(weekend = day_of_week %in% c("Sat", "Sun"))|> ggplot() + geom_col(aes(x=day_of_week, y=mean_births, fill = weekend)) )`
2. `births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) ggplot() |> geom_col(aes(x = day_of_week, y = mean_births, fill = weekend)`
3. `births |> group_by(day_of_week) |> summarize (mean_births = mean(births)) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x=day_of_week, y = mean_births), fill = weekend) + labs(x = "Day of Week", y = "Avg. births")`
4. `births |> group_by(day_of_week) |> summarize(mean_birth = mean(births)) |> ggplot()+geom_col(aes(x = day_of_week,y=mean_birth))`
5. `births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) ggplot() + geom_col(aes(x = day_of_week, y= mean_births, fill = weekend) ) + labels(x="Day of week", y= "Avg. births")`

# Student responses

---

1. `births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x=day_of_week, y=mean_births, fill=weekend)) + labs(x = "Day of week", y = "Average number of births per day")`
2. `births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x = day_of_week, y = mean_births, fill = weekend)) + labs(x = "Day of the Week", y = "Average Number of Births per Day")`
3. `births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x=day_of_week, y=mean_births, fill = weekend)) + labs(x="Days of week", y="Average birthdays")`
4. `births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> print() ggplot()+geom_col(aes(x= day_of_week, y = mean_births))`

# Student responses

---

1. `births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x = day_of_week, y = mean_births, fill = weekend)) |> lab(x = "Day of Week", y = "AVG Births")`
2. `births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> ggplot() + geom_col(aes(x = day_of_week, y = mean_births)) + labs(x = "Day of Week", y = "Number of Births")`
3. `births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> ggplot() + geom_col(aes(x= day_of_week, y = mean_births))+ labs(x="Day of week",y="average number of births")`
4. `births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x=day_of_week, y=mean_births, fill = weekend)) + labs(x = "Day of week", y = "Avg # Births")`

# Student responses

---

1. `births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> mutate(weekend = day_of_week %in% c("Sat","Sun")) |> ggplot() +  
geom_col(aes(x=day_of_week, y = mean_births, fill = weekend))`
2. `births |> group_by(day_of_week)|> summarize(mean_births=mean(births))`
3. `births |> group_by(day_of_week)|> summarize(mean_births = mean(births)) |> mutate(weekend = day_of_week %in% c("sat", "sun")) |> ggplot()+ geom_col(  
aes(x = day_of_week, y = mean_births, fill = weekend)) labs(x = "Day of week", y = "Avg. births")`
4. `births |> group_by(day_of_week)|> summarize(mean_births = mean(births))|> mutate(weekend = day_of_week %in% c("Sat","Sun"))|> ggplot()+geom_col(  
aes(x=day_of_week, y = mean_births,fill = weekend) )+ labs( x="Day of week", y= "Avg. births" )`
5. `births |> group_by(day_of_week)|> summarize(mean_births =mean(births))|> mutate(weekend=day_of_week %in% c("Sat", "Sun"))|>  
ggplot()+geom_col(aes(x=day_of_week, y = mean_births, fill = weekend))+ labs(x="day of week", y="Ave births")`



# Student responses

---

1. `births |> head()` `grouped_births <- births |> group_by(day_of_week) |> summarize(mean_births = mean(births))` `ggplot(grouped_births, aes(x = day_of_week, y = mean_births)) + geom_col() + labs(x = "Day of week", y = "avg. number of birthday per day")`
2. `births |> head()` `weekday_births <- births %>% group_by(day_of_week) %>% summarize(mean_births = mean(births, na.rm = TRUE))` `ggplot(weekday_births, aes(x = day_of_week, y = mean_births)) + geom_col() + labs( x = "Day of week", y = "Average number of births per day" )`
3. `births |> head()` `weekday_births <- births |> group_by(day_of_week) |> summarise(mean_births = mean(births))` `weekday_births |> ggplot(aes(x = day_of_week, y = mean_births)) + geom_col() + labs( x = "Day of the Week", y = "Average Number of Births" )`
4. `births |> head()` `weekday_births <- births |> group_by(day_of_week) |> summarize(mean_births = mean(births))` `weekday_births |> ggplot(aes(x = day_of_week, y = mean_births)) + geom_col() + labs( x = "Weekday", y = "Average number of births per day" )`

# Student responses

---

1. `births |> head()` `weekday_births <- births |> group_by(day_of_week) |> summarize(mean_births = mean(births))` `weekday_births |> ggplot(aes(x = day_of_week, y = mean_births)) + geom_col(fill = "lightblue") + labs( title = "Average Births by Day of the Week", x = "Day of the Week", y = "Average Number of Births per Day" )`
2. `births |> head()` `|> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> ggplot() + geom_col(aes(x = day_of_week, y = mean_births))`
3. `births|> group_by(day_of_week)|> summarize(mean_births = mean(births))|> ggplot()+ geom_col(aes(x=day_of_week, y = mean_births))`
4. `library(dplyr) library(ggplot2) weekday_births <- births |> group_by(day_of_week) |> summarise( mean_births = mean(births, na.rm = TRUE), .groups = "drop" ) ggplot(weekday_births, aes(x = day_of_week, y = mean_births)) + geom_col(fill = "steelblue") + labs( x = "Day of the week", y = "Average number of births per day" ) + theme_minimal()`

# Student responses

---

1. `weekday_births <- births %>% group_by(day_of_week) %>% summarize(mean_births = mean(births, na.rm = TRUE)) ggplot(weekday_births, aes(x = day_of_week, y = mean_births)) + geom_col() + labs( x = "Weekday", y = "Average number of births per day" )`
2. `weekday_births <- births |> group_by(day_of_week) |> summarize( mean_births = mean(births) ) ggplot(weekday_births, aes( x = day_of_week, y = mean_births ) ) + geom_col() + labs( x = "Day of week", y = "Average births per day", title = "Average births by weekday" )`
3. `weekday_births <- births |> group_by(day_of_week) |> summarize(mean_births = mean(births, na.rm = TRUE)) weekday_births ggplot(aes(x = day_of_week, y = mean_births)) + geom_col() + labs( x = "Day of Week", y = "Average Number of Births" )`

# Student responses

---

1. `weekday_births <- births |> group_by(day_of_week) |> summarize(mean_births = mean(births), .groups = "drop") weekday_births |> ggplot(aes(x = day_of_week, y = mean_births)) + geom_col() + labs( title = "Average births by day of week", x = "Day of week", y = "Average births per day" )`
2. `weekday_births <- births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) ggplot(weekday_births) + geom_col(aes(x = day_of_week, y = mean_births, fill = weekend)) + labs( x = "Day of week", y = "Average number of births per day" )`
3. `weekday_births <- births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) ggplot(weekday_births, aes(x = day_of_week, y = mean_births, fill = weekend)) + geom_col() + labs( x = "Weekday", y = "Average number of births per day" )`

# Student responses

---

1. `weekday_births <- births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> mutate(weekend = day_of_week %in% c("Sat", "Sun"))  
weekday_births |> ggplot() + geom_col(aes(x = day_of_week, y = mean_births, fill = weekend)) + labs(x = "Day of week", y = "Average births per day")`
2. `weekday_births <- births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |>  
ggplot() + geom_col(aes(x = day_of_week, y = mean_births, fill = weekend)) + labs(x = "Day of week", y = "Avg. births") weekday_births`
3. `weekday_births <- births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) ggplot(weekday_births, aes(x = day_of_week, y =  
mean_births)) + geom_col() + labs(x = "Day of week", y = "Average births per day")`
4. `weekday_births <- births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> ggplot() + geom_col(aes(x = day_of_week, y =  
mean_births)) + labs(title = "Distribution of birth days", x = "Weekday", y = "Births per day")`

# Student responses

---

1. `weekday_births <- births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> mutate(weekend = day_of_week %in% c("Sat", "Sun"))  
weekday_births |> ggplot() + geom_col(aes(x = day_of_week, y = mean_births, fill = weekend)) + labs(x = "Day of Week", y = "Mean Births")`
2. `weekday_births <- births |> group_by(day_of_week) |> summarize( mean_births = mean(births) ) |> ggplot() + geom_col(aes(x = day_of_week, y =  
mean_births)) + labs( x = "Day of week", y = "Average births per day", title = "births per week day")`
3. `weekday_births <- births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) ggplot() + geom_col( aes(x = day_of_week, y = mean_births))`
4. `weekday_births <- births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) weekday_births |> ggplot() + geom_col(aes(x=day_of_week,  
y=mean_births)) + labs(x = 'Weekday', y = 'Average Number of Births Per Day')`

# Student responses

---

1. `weekday_births <- births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> slice_head(n = 7) ggplot(weekday_births, aes(x = day_of_week, y = mean_births)) + geom_col() + labs( x = "Day of Week", y = "Average Births per Day" )`
2. `weekday_births <- births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x = day_of_week, y = mean_births, fill = weekend)) + labs(x = "Day of week",y = "Avg. births")`
3. `weekday_births<- births|> group_by(day_of_week)|> summarize(mean_births=mean(births)) weekday_births |> ggplot() + geom_col(aes(x = day_of_week, y = mean_births)) + labs( x = "Day of week", y = "Average number of births per day")`
4. `weekday_births<-births |> group_by(day_of_week)|> summarize(mean_births=mean(births)) weekday_births |> ggplot() + geom_col(aes(x = day_of_week, y = mean_births)) + labs( title = "Average Births by Day of the Week", x = "Day of the Week", y = "Mean Number of Births" )`

# Student responses

---

1. `weekday_births<-births|> group_by(day_of_week)|> summarize(mean_births = mean(births))|> mutate(weekend = day_of_week %in% c("Sat", "Sun"))|> ggplot()+ geom_col(aes(x=day_of_week, y = mean_births)) labs( x = "Day of week", y = "Average births per day", fill = "Weekend" ) + scale_fill_manual(values = c("FALSE" = "steelblue", "TRUE" = "tomato"))`
2. `weekly_births <- births |> group_by(day_of_week) |> summarize(mean_births = mean(births)) |> mutate(weekend = day_of_week %in% "Sat", "Sun") |> mutate(day_type = ifelse(day_of_week %in% c("Saturday", "Sunday"), "Weekend", "Weekday")) |> ggplot(aes(x = day_of_week, y = mean_births, fill = weekend)) + geom_col() + labs( x = "Day of week", y = "Average births per day", fill = "" ) weekly_births`



**What would you expect July 4 births to look like across years if births are less common on certain weekdays? What would the pattern look like if births were less common on holidays as well as weekends?**

---

0 anonymous responses

No responses were submitted.

**- Filter `births` to July 4 in every year and save the result. - Make a bar plot with `year` on the x-axis and `births` on the y-axis. - Color each bar by its `day\_of\_week`. - Label the axes and give the plot a title.**

---

74 anonymous responses

Other responses

74

# Student responses

---

1. births
2. births
3. births
4. `births` `july4_births <- births %>% filter(month == 7, day == 4) ggplot(july4_births, aes(x = factor(year), y = births, fill = day_of_week)) + geom_col() + labs(x = "Year", y = "Number of births", title = "Births on July 4 across years" )`
5. `births` `july4 <- births |> filter(month == 7, day == 4) july4 |> ggplot(aes(x = year, y = births, fill = day_of_week)) + geom_col() + labs(x = "Year", y = "Number of births", title = "July 4 Births Across Years" )`
6. `births` `july4_births <- births |> filter(month == 7, date_of_month == 4) july4_births |> ggplot(aes(x = factor(year), y = births, fill = day_of_week)) + geom_col() + labs(title = "July 4 Births Across Years", x = "Year", y = "Number of Births", fill = "Day of the Week" )`
7. `births |> filter(month == 7 , date_of_month == 4) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col( aes(x = year, y = births, fill = day_of_week))`
8. `births |> filter(month == 7, date_of_month == 4)`

# Student responses

---

1. `births |> filter(month == 7, date_of_month == 4) |> ggplot(aes(x = year, y = births, fill = day_of_week)) + geom_col() + labs(x = "Year", y = "Number of births", fill = "Day of week", title = "Births on July 4 across years" )`
2. `births |> filter(month == 7, date_of_month == 4) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes (x = year, y = births, fill = weekend)) + labs(x = "year", y = "births")`
3. `births |> filter(month == 7, date_of_month == 4) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x = year, y = births, fill = weekend))`
4. `births |> filter(month == 7, date_of_month == 4) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x = year, y = births, fill = weekend))`
5. `births |> filter(month == 7, date_of_month == 4) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x = year, y = births, fill = day_of_week)) + labs(x = "Year", y = "Number of Births", title = "Title")`

# Student responses

---

1. `births |> filter(month == 7, date_of_month == 4) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x = year, y = births, fill = weekend))`
2. `births |> filter(month == 7, date_of_month == 4) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot(aes(x = year, y = births, fill = weekend)) + geom_col() + labs( x = "Year", y = "Number of births", title = "July 4 births by year", fill = "Weekend" )`
3. `births |> filter(month == 7, date_of_month ==4) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x = year, y = births, fill = weekend))`
4. `births |> filter(month == 7 & date_of_month == 7) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x = year, y = births, fill = weekend))`
5. `births |> filter(month == 7, date_of_month == 4) |> mutate (weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x= year, y = births, fill = weekend))`

# Student responses

---

1. `births |> filter(month == 7, date_of_month == 4) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x=year, y = births, fill = weekend))`
2. `births |> filter(month == 7, date_of_month == 4) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x = year, y = births, fill = weekend))`
3. `births |> filter(month == 7, date_of_month == 4) |> mutate(weekend = day_of_week %in% c("Sat", "Sun", "Mon")) |> ggplot() + geom_col(aes(x= year, y = births, fill = weekend))`
4. `births |> filter(month == 7, date_of_month == 4) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot + geom_col(aes(x=yea, y=births, fill=weekend)) + labs(x = "Day of week", y = "Average number of births per day")`
5. `births |> filter(month == 7, date_of_month == 4) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x = year, y = births, fill = weekend))`
6. `births |> filter(month == 7, date_of_month == 4) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x = year, y = births, fill = weekend))`

# Student responses

---

1. `births |> filter(month == 7, date_of_month == 4) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x = year, y = births, fill = weekend))`
2. `births |> filter(month == 7, date_of_month == 4) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x = year, y = births, fill = weekend))`
3. `births |> filter(month == 7, date_of_month == 4) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x = year, y = births, fill = weekend)) + labs (x = "Year", y = "Number of Births", title = "July 4 Births across Years")`
4. `births |> filter(month == 7, date_of_month == 4) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x = year, y = births, fill = weekend)) + labs( title = "Day of week vs births" )`
5. `births |> filter(month == 7, date_of_month == 4) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x=year, y=births, fill=weekend))`

# Student responses

---

1. `births |> filter(month == 7, date_of_month == 4) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x=year,y=births, fill=weekend))`
2. `births |> filter(month == 7, date_of_month == 4) |> mutate(weekend = day_of_week %in% c("Sat", "Sun"))|> ggplot() + geom_col(aes(x=year, y =births, fill = weekend))`
3. `births |> filter(month ==7, date_of_month ==4) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x = year, y = births, fill = weekend))`
4. `births |> filter(month == 7, date_of_month == 4) |> ggplot() + geom_col(aes(x = year, y = births, fill = day_of_week)) + labs(title = "July 4th births across the years", x = "Year", y = "Births")`
5. `births |> filter (month == 7, date_of_month == 4) |> mutate(weekend = day_of_week%in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x = year, y= births, fill = weekend))`
6. `births |> filter(month == 7, date_of_month == 4) | mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x = year, y = births, fill = weekend))`



# Student responses

---

1. `births |> filter(month == 7, date_of_month == 4) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x = year, y = births, fill = weekend))`
2. `births |> filter(month == 7, date_of_month == 4) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x = year, y = births, fill = weekend))`
3. `births |> filter(month == 7, date_of_month == 4) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x = year, y = births, fill = weekend)) + labs(x = "Year", y = "Births", main = "Births on July 4 per Year")`
4. `births |> filter(month == 7, date_of_month == 4) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> print() ggplot() + geom_col(x = year, y= births, fill = weekend)`
5. `births |> filter(month == 7, date_of_month == 4) |> mutate(weekdend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x = year, y = births, fill = weekend))`

# Student responses

---

1. `births |> filter(month == 7, date_of_month == 4) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x = year, y = births, fill = weekend))`
2. `births |> filter(month == 7, date_of_month == 4) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) ggplot() + geom_col(aes(x = year, y = births, fill = weekend))`
3. `births |> filter(month == 7, date_of_month == 4) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x = year, y = births, fill = weekend)) labs(title = "July 4 Births", x = "Births", y = "Year")`
4. `births |> filter(month == 7, date_of_month == 4) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x = year, y = births, fill = day_of_week))`
5. `births |> filter(month == 7, date_of_month == 4) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x = year, y = births, fill = weekends))`

# Student responses

---

1. `births |> filter(month == 7, date_of_month == 4) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x = year, y = births, fill = weekend))`
2. `births |> filter(month == 7, date_of_month == 4) |> ggplot() + geom_col(aes(x = year, y = births))`
3. `births |> filter(month == 7, date_of_month == 4) |> ggplot(aes(x = year, y = births, fill = day_of_week)) + geom_col() + labs( x = "Year", y = "Births", title = "July 4 Births Across Years" )`
4. `births |> filter(month == 7, date_of_month == 4) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x = year, y = births, fill = weekend))`
5. `births |> filter(month == 7, date_of_month == 4) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x = year, y = births, fill = weekend))`
6. `births |> filter(month == 7, date_of_month == 4) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x=year, y=births, fill = weekend))`

# Student responses

---

1. `births |> filter(month == 7, date_of_month == 4) |> mutate(weekend = day_of_week %in% c("Sat","Sun")) |> ggplot() +  
 geom_col(aes(x=year,y=births,fill=weekend))`
2. `births |> filter(month == 7, date_of_month == 4) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x = year, y = births, fill =  
 weekend))`
3. `births |> filter(month == 7, date_of_month == 4)|> mutate(weekend = day_of_week %in% c("Sat", "Sun"))|> ggplot()+geom_col(aes(x=year, y=births,  
 fill=weekend))`
4. `births |> filter(month == 7, date_of_month ==4) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes (x = year, y = births, fill =  
 weekend))`
5. `births |> filter(month == 7, date_of_month ==4) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x = year, y = births, fill =  
 weekend)) + labs(title = "Births on July 4th", x = "Year", y = "Birth Count")`
6. `births |> filter(month == 7,date_of_month == 4)`

# Student responses

---

1. `births |> filter(month == 7,date_of_month == 4) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x = year, y = births, fill = weekend))`
2. `births |> filter(month ==7, date_of_month == 4) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x=year,y=births,fill=weekend))`
3. `births |> filter(month ==7, date_of_month ==4) |> mutate(weekend = day_of_week % in % c("sat","sun")) |> ggplot(aes(x= year, y= births, fill = weekend))`
4. `births |> filter(month ==7, day_of_month == 4 )|> mutate(weekend = day_of_week %in% c("Sat", "Sun"))|> ggplot()+geom_col(aes(x= year, y =births, fill = weekend))`
5. `births|> filter(month == 7, date_of_month == 4)|> mutate(weekend =day_of_week %in% c("Sat", "Sun"))|> ggplot() + geom_col(aes(x = year, y = births, fill = weekend))`
6. `births|> filter(month==7, date_of_month==4)|> mutate(weekend = day_of_week %in% c("Sat","Sun"))|> ggplot()+ geom_co,(aes(x=year, y=births, fill=weekend))`

# Student responses

---

1. `births|> filter(month==7, date_of_month==4)|> mutate(weekend=day_of_week %in% c("Sat","Sun"))|> ggplot()+geom_col(aes(x=year,y=births, fill=weekend))`
2. `jul4 <- births |> filter(month == 7 & date_of_month == 4) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) jul4 |> ggplot() + geom_col(aes(x=year, y=births, fill = day_of_week))`
3. `july_4 <- births |> filter(month == 7 & date_of_month == 4) ggplot(july_4, aes(x = year, y = births, fill = day_of_week)) + geom_col() + labs( x = "Year", y = "Births", title = "Births on July 4 by Year" )`
4. `july4 <- births |> filter(month == 7, date_of_month == 4) july4 |> ggplot(aes(x = year, y = births, fill = day_of_week)) + geom_col() + labs( title = "July 4 births by year", x = "Year", y = "Births", fill = "Day of week" )`
5. `july4_births <- births %>% filter(month == 7, day == 4) ggplot(july4_births, aes(x = factor(year), y = births, fill = day_of_week)) + geom_col() + labs( x = "Year", y = "Number of births", title = "Births on July 4 across years" )`

# Student responses

---

1. `july4_births <- births |> filter(month == 7, date_of_month == 4) ggplot(july4_births, aes(x = year, y = births, fill = day_of_week)) + geom_col() + labs( x = "Year", y = "Births", title = "Number of births on July 4 by year" )`
2. `july4_births <- births |> filter(month == 7, date_of_month == 4) july4_births |> ggplot(aes(x = year, y = births, fill = day_of_week)) + geom_col() + labs( x = "Year", y = "Number of Births", title = "Births on July 4 by Year and Weekday" )`
3. `july4_births <- births |> filter(month == 7, date_of_month == 4) ggplot(july4_births, aes(x = year, y = births, fill = day_of_week)) + geom_col() + labs( x = "Year", y = "Births", title = "Number of births on July 4" )`
4. `july4bdays <- births |> filter(month == 7, date_of_month == 4) |> mutate(weekend = day_of_week %in% c("Sat", "Sun")) |> ggplot() + geom_col(aes(x = years, y = births, fill = weekend))`

# Student responses

---

1. `july4birth <- births |> filter(month == 7 & date_of_month == 4) ggplot(july4birth, aes(x = year, y = births, fill = day_of_week)) + geom_col() + labs( x = "Year", y = "Births", title = "Births on July 4 Across Years" )`
2. `library(dplyr) library(ggplot2) july4_births <- births |> filter(month == 7, day_of_month == 4) ggplot(july4_births, aes(x = year, y = births, fill = factor(day_of_week))) + geom_col() + labs( title = "Births on July 4 by year", x = "Year", y = "Number of births", fill = "Day of the week" ) + theme_minimal()`



# Would you expect more or fewer births on dates like these? What comparison would show whether relatively more or fewer births occur on these dates?

---

82 anonymous responses

## Choice/celebration motivates more births

23

Asserts that people would deliberately time births or deliveries to obtain novel or desirable dates, increasing birth counts on those days.

## Believes no correlation with dates

15

Argues that birth timing is not under calendar control, so there should be no relationship between these special dates and birth counts.

## Assumes slight increase

12

Expresses a belief that births are somewhat more frequent on these dates but expects the effect to be small.

## Brief assertion of increase

8

Short, unelaborated statements claiming there are more births on these dates without justification or method.

# Response themes for question 11

---

## Compare special date to surrounding week

6

Suggests comparing the special date's birth count to the week around it to detect relative differences.

---

## Expect fewer births on some dates

6

Claims certain special dates will have fewer births rather than more.

---

## Suggests statistical comparison method

6

Proposes using a formal statistical test (comparing means across days) to determine if these dates have relatively more births.

---

## Use simple visuals to compare days

6

Recommends basic data visualization, like a bar chart, to compare birth counts across dates.

---

# Student responses

---

1. I don't expect there to be a super drastic change between these different days, but I guess maybe that there would be more births on these days because humans are really goofy and honestly would do whatever they want so I think there is a pretty good potential for births to be larger here
2. I expect births in these days to be higher than the normal days, considering people's preferences to make them unique
3. I expect more people want to have birthdays/births on this day. I expect this because it is a fun birthday to have, so many mothers would push their birth to be on that specific day
4. I expect there to be more births on these days because people want to try and make the event more memorable and for some reason they decide this
5. I expect there would be more births occurring on these dates.
6. I expect to be more around special days
7. I might expect more scheduled births on memorable repeated dates because parents may prefer a distinctive birthday.
8. I think it would be more popular since people love to talk about birth dates

# Student responses

---

1. I think people on this have lot more birth because they believe in the angel number. I also think that since it only assumes these two dates we could check if it is better
2. I think people will usually try to aim for these days, given theyre not on a weekend or on a holiday. The actually day would be a pretty normal day, but the date would be special
3. I think these type of dates would be more popular because they seem "cooler" to some people, given there is some flexity of when a family can have a child
4. I think they would be more popular because people want their kids to be special
5. I would expect memorable days to be more popular. It's literally a once in a lifetime chance and it's cool to have
6. i would expect more because if given the choice, people would want to have their kids on these days
7. I would expect more because maybe moms are doing things that can speed up labor to end up giving birth on one of these days so that their kids have a cool birthday.
8. I would expect more birth days on these dates because people may aim for the uniqueness of the birthday, and affect their planning

# Student responses

---

1. I would expect more births on these days so there would be a relatively higher amount of births
2. I would expect that people would want to have children on memorable days because they are easier to remember and they might want to brag about how their child has a cool birthday.
3. I would expect these kinds of days to be more popular, as special days seem more interesting of a birthday.
4. More for sure. People would aim for them
5. More popular, it is cool to have a child with a unique once ever existing birth date.
6. These dates would be more popular because if people were aiming to give birth on these dates, there would generally be more births.
7. We might expect more births on these days since they are very interesting and special dates so people might have waited so that their child could be born specifically on this day

# Student responses

---

1. fewer because so random
2. I do not expect these days to be more or less popular. I feel like they are not that memorable that so much work would go in to making sure their child was born on these days
3. I feel like a memorable date would not matter. You can't really control when the baby comes and I doubt memorable days have more births.
4. I think it would be about the same because it's not that big of a deal to have a birthday like that.
5. I think it would be the same because I dont think people are thinking that much about it
6. I would expect the same amount of births on these days, as I expect that there are a lot of factors that go into deciding when a child is born, so any influence of trying to have a "memorable" birthday would be small.
7. I would not expect any difference, because I don't think people would go out of there way to do this/
8. I wouldn't expect more birthdays to occur on these dates because there isn't something pushing mothers to give birth just because the date is memorable. Compare these to the dates around it.

# Student responses

---

1. I wouldn't expect the memorable birthdays to be correlated, just because it's hard to aim for a specific day. We would have to compare the avg births on these memorables dates vs the avg births for the rest of that month.
2. It is impossible to actually know, but likely no sign of being more. The dates are purely arbitrary.
3. No correlation because can't choose when the baby comes
4. No difference, I don't think its possible to predict when your child is going to be born.
5. should be same since every day is random
6. The exact same amount of birthdays. o r actually less
7. THe same, as we cannot control when pepopel are born

# Student responses

---

1. I believe that there would be relatively little change in the number of births on these dates in the specific years given. There may be a small increase due to people trying to "aim" for these days, but too many other factors are at play for this to be a major cause of difference.
2. I expect it to be more popular but not that significant. People may aim for these days but for medically controlled birth that is not a strong factor.
3. I think that there may be slightly more births on those days as parents could want their kids to have a special birthday for a special boy or girl
4. I think that there will be slightly more, but that the weekend effect would override it. You could compare them to similar days across the years, ie graph 7/7 births for each year
5. I would assume more, but not by a lot. We could compare 07/07 across years and see if it increased noticeably in 2007. We could also compare 07/07 to the couple days before and after it.
6. I would expect these days to be slightly more frequent than others.



# Student responses

---

1. I would expect these days to be slightly more popular because like we saw in the runners when people have a goal they will likely be distributed close to it so if people are going to give birth around that time and want their kids birthday to be that day they will likely try and make that happen. But i do expect this to be a very small difference
2. slightly more on dates like those, right comparison is each memorable date against neighboring dates,
3. they could be slightly more popular if people were scheduling cesarean sections and such for them, but I wouldn't think there would be a huge difference
4. they may be more popular but i dont think itll be a significant differnce
5. They might be slightly more popular, but I do not expect a notable increase of births on these days
6. Yes, you would expect them to be slightly more popular but less significant than the day of week or if it is a holiday

# Student responses

---

1. 07will be more births
2. I expect more births on these days
3. I would expect more births on memorable dates. to compare we can pick a date nearby to a memorable and see the difference in # of births
4. I would expect more births because they want to do it before special holidays
5. more I guess, because the way we group it
6. More popular
7. MORE, relatively more
8. Prolly more

# Student responses

---

1. I would expect more births on memorable dates like 07/07/07 or 12/12/12 because some parents may choose these dates for scheduled births. I would compare the number of births on these memorable dates with nearby dates, while also considering the day of the week.
2. I would expect more births on those special dates. For comparison, I will first define and classify the special dates and then make a bar plot on the average number of births for each of the category.
3. I would expect more, because a memorable birthday is desirable, and a comparison between the special date and the week surround it would show if relatively more births occur
4. If people really were trying to get these days, I would expect more births on these days but only comparative to the rest of the year as I suspect the days around them might also be very high due to births taking an inconsistent amount of time. Maybe also not many births the day before?
5. More because they are more memorable. You could compare them to other days like other july 7ths
6. Yes. The frequency of the date of birth on exact that day should be very high

# Student responses

---

1. i expect most of these dates to have more births, while some, like 6/6/6, will have less.
2. i think there are less births on these days
3. I would expect fewer births on these dates because people wouldnt not want them born on holidays. You cna compare of this is true by seeing the jump in borths wintin 1 or 2 days of the holidays
4. I would expect these birthdays to be less popular because people are busier on holidays and wouldn't plan c sections on these days. We could compare mean births on these set holidays to mean births on not on holidays and see if there is a difference.
5. I would expect these days to be slightly less popular because people are busy but it is hard to plan a day to give birth.
6. The special days may have the slightly higher births, but since people now are less interested in giving birth on the special day, the births should be return to normal.

# Student responses

---

1. I think not because this date seems to have no affect on birth. But if we want to compare, we could use t test on one of the common days need this special date and the special date itself, keeping the day\_of\_week same.
2. I would expect more births on these dates. You could take the mean births of every dec 12 and the compare it to dec 12, 2012.
3. i would expect more births. compare this day to the day before
4. More births; t-test for means between every other day and these days
5. probabaly 12/12/12? cause as the population growth. More birth would be given as the time goes
6. Think its should be same with average, no one can control the birth, maybe use zscore to check

# Student responses

---

1. Bar chart would show this data. I would expect it to be similar in popularity to any typical day of the week.
2. compare the average births on those days with rest of year to see if it is more popular or not
3. I feel like the trend for these days would follow a similar trend, where it seems to be up on during the week, and down on the weekend. Comparing to nearby weeks would be a good comparison.
4. I would expect more births on memorable dates because some parents may choose to schedule deliveries on those days. To test this, I would compare the number of births on memorable dates with nearby dates that fall on the same day of the week.
5. I would expect these days to be more popular. We could compare births on the same day across years to see if a certain year stood out where the month and day matched.
6. Maybe this could be true for a year where it was a popular idea.

- Sort `births` by `year`, `month`, and `date\_of\_month`. - Before filtering, create `before = lag(births, 7)` and `after = lead(births, 7)`. - Keep rows where `month`, `date\_of\_month`, and `year - 2000` match; save as `repeat\_comparison`. - Display `year`, `month`, `date\_of\_month`, `before`, `births`, and `after`.

---

78 anonymous responses

Other responses

78

# Student responses

---

1. `births |> arrange(year, month, date_of_month) |> mutate( before = lag(births, 7), after = lead(births, 7), yearm2k = year - 2000 ) |> filter( date_of_month == month, yearm2k == month )`
2. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7) |> yearm2k = year -2000) |> filter(date_of_month == month, yearm2k == month)`
3. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7), yearm2k = year - 2000) |> filter(date_of_month == month, yearm2k == month)`
4. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7), yearm2k = year - 2000) |> filter(date_of_month == month, yearm2k == month)`
5. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7), year2k = year - 2000) |> filter(date_of_month == month, year2k == month)`



# Student responses

---

1. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7), yearm2k = year - 2000) |> filter(date_of_month == 7, yearm2k == month)`
2. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7), yearm2k = year - 2000) |> filter(date_of_month == month, yearm2k == month) |>`
3. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7), yearm2k = year - 2000) |> filter(date_of_month == month, yearm2k == month)`
4. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7), yearm2k = year - 2000) |> filter(date_of_month == month, yearm2k == month) |> select(year, month, date_of_month, before, births, after)`

# Student responses

---

1. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7), yearm2k = year - 2000) |> filter(date_of_month == month, yearm2k == month)`
2. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7), yearm2k = year - 2000) |> filter(date_of_month == month, yearm2k == month)`
3. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7), yearm2k = year - 2000) |> filter(date_of_month == month, yearm2k == month)`
4. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7), yearm2k = year - 2000) |> filter(date_of_month == month, yearm2k == month)`
5. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7), yearm2k = year - 2000) |> filter(month == date_of_month, yearm2k == month) |> select(year, month, date_of_month, before, births, after)`

# Student responses

---

1. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7), yearm2k = year - 2000) |> filter(date_of_month == month, yearm2k == month) #head()`
2. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7), yearm2k = year - 2000) |> filter(date_of_month == month, yearm2k == month)|> select(year, month, date_of_month, before, births, after)`
3. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7), yearm2k = year - 2000) |> filter(date_of_month == month, yearm2k == month)`
4. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7), yearm2k = year - 2000) |> filter(date_of_month == month, yearm2k == month)`

# Student responses

---

1. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7), yearm2k = year - 2000) |> filter(date_of_month == month, yearm2k == month)`
2. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7), yearm2k = year - 2000) |> filter(date_of_month == month, yearm2k == month)`
3. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7), yearm2k = year - 2000) |> filter(date_of_month == month & date_of_month == yearm2k)`
4. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7)) |> filter(date_of_month == month)`
5. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7)) |> mutate(yearm2k = year - 2000) |> filter(date_of_month == month, yearm2k == month)`

# Student responses

---

1. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7)) |> filter(date_of_month == month, month == year - 2000) |> select(year, month, date_of_month, before, births, after)`
2. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7), yearm2k = year - 2000) |> filter(date_of_month == month, yearm2k == month)`
3. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7), year2k = year - 2000) |> filter(date_of_month == month, month == year2k) |> slice_sample(n = 10)`
4. `births |> arrange(year, month, date_of_month)|> mutate(before = lag(births,7), after = lead(births,7), yearm2k = year-2000) |> filter(date_of_month == month, yearm2k == month)|> slice_sample(n=10)`
5. `births |> arrange(year, month, date_of_month)|> mutate(before=lag(births,7), after=lead(births,7), yearm2k=year-2000)|> filter(date_of_month==month, yearm2k==month)->repeat_comparison`

# Student responses

---

1. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7), yearm2k = year - 2000) |> filter(date_of_month == month, yearm2k == month)`
2. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7), yearm2k = year - 2000) |> filter(date_of_month == month) |> slice_sample(n = 10)`
3. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7), yearm2k = year - 2000) |> filter(year - 2000 == month, month == date_of_month) |> select(year, month, date_of_month, before, births, after)`
4. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7), yearm2k = year - 2000) |> filter(date_of_month == month, yearm2k == month)`

# Student responses

---

1. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7), yearm2k = year-2000) |> filter(date_of_month == month, yearm2k == month)`
2. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7), yearm2k = year - 2000) |> filter(date_of_month == month, yearm2k == month)`
3. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births,7), after = lead(births,7), yearm2k = year - 2000) |> filter(date_of_month == month, yearm2k == month)`
4. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births,7), after = lead(births, 7), yearm2k = year- 2000) |> filter(date_of_month == month, yearm2k == month) |>`
5. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7), yearm2k = year -2000) |> filter(date_of_month == month, yearm2k == month)`

# Student responses

---

1. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7), yearm2k = year - 2000) |> filter(date_of_month == month, yearm2k == month)`
2. `births |> arrange(year, month, date_of_month) |> mutate( before = lag(births, 7), after = lead(births, 7) ) |> filter( month == date_of_month, month == year-2000 ) -> repeat_comparison repeat_comparison`
3. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lag(births, 7), yearm2k = year - 2000) |> filter(date_of_month == month) |> slice_sample(n=10)`
4. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7), yearm2k = year - 2000) |> filter(date_of_month == month) |> slice_sample(n=10)`
5. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7), yearm2k = year - 2000) |> filter(date_of_month == month, yearm2k == month)`



# Student responses

---

1. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7), yearm2k = year - 2000) |> filter(date_of_month == month, yearm2k == month)`
2. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7), yearm2k = year - 2000) |> filter(date_of_month == month, yearm2k == month)`
3. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7), yearm2k = year - 2000) |> filter(date_of_month == month, month == yearm2k)`
4. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7), yearm2k = year - 2000) |> filter(date_of_month == month, yearm2k == month) |> slice_sample(n=10)`
5. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7), yearm2k = year - 2000) |> filter(date_of_month == month, yearm2k == month)`

# Student responses

---

1. `births |> arrange(year, month, date_of_month) |> mutate(before = lag(births,7), after = lead(births, 7), yearm2k = year - 2000) |> filter(date_of_month == month, yearm2k == month)`
2. `births |> arrange(year,month,date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7), yearm2k = year - 2000) |> filter(date_of_month == month, yearm2k == month)`
3. `births |> head repeat_comparison <- births |> arrange(year, month, date_of_month) |> mutate( before = lag(births, 7), after = lead(births, 7) ) |> filter( month == date_of_month, date_of_month == year - 2000 ) repeat_comparison |> select(year, month, date_of_month, before, births, after)`
4. `births |> head()`
5. `births |> head() arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7), yearm2k = year - 2000) |> filter(date_of_month == month, yearm2k == month)`

# Student responses

---

1. `births |> head() repeat_comparison <- births |> arrange(year, month, date_of_month) |> mutate( before = lag(births, 7), after = lead(births, 7) ) |> filter(month == date_of_month & date_of_month == year - 2000) repeat_comparison |> select(year, month, date_of_month, before, births, after)`
2. `births |> head() repeat_comparison <- births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7)) |> filter(month == date_of_month & date_of_month == year - 2000) |> select(year, month, date_of_month, before, births, after) repeat_comparison`
3. `births |> head() repeat_comparison <- births_df |> arrange(year, month, date_of_month) |> mutate( before = lag(births, 7), after = lead(births, 7) ) |> filter(month == date_of_month, year - 2000 == month) |> select(year, month, date_of_month, before, births, after) repeat_comparison`

# Student responses

---

1. `library(dplyr) repeat_comparison <- births |> arrange(year, month, date_of_month) |> mutate( before = lag(births, 7), after = lead(births, 7) ) |> filter(month == date_of_month, month == year - 2000) |> select(year, month, date_of_month, before, births, after) repeat_comparison`
2. `repeat_comparison <- births |> arrange(year, month, date_of_month) |> mutate( before = lag(births, 7), after = lead(births, 7) ) |> filter( month == date_of_month, month == year - 2000 ) |> select(year, month, date_of_month, before, births, after)`
3. `repeat_comparison <- births |> arrange(year, month, date_of_month) |> mutate( before = lag(births, 7), after = lead(births, 7) ) |> filter( month == year - 2000, date_of_month == year - 2000 ) |> select(year, month, date_of_month, before, births, after) repeat_comparison`

# Student responses

---

1. `repeat_comparison <- births |> arrange(year, month, date_of_month) |> mutate( before = lag(births, 7), after = lead(births, 7) ) |> filter(month == date_of_month, date_of_month == year - 2000) |> select(year, month, date_of_month, before, births, after) repeat_comparison`
2. `repeat_comparison <- births |> arrange(year, month, date_of_month) |> mutate( before = lag(births, 7), after = lead(births, 7) ) |> filter(month == date_of_month, year - 2000 == month) repeat_comparison |> select(year, month, date_of_month, before, births, after)`
3. `repeat_comparison <- births |> arrange(year, month, date_of_month) |> mutate( before = lag(births, 7), after = lead(births, 7) ) |> filter(month == date_of_month, year - 2000 == month) |> select(year, month, date_of_month, before, births, after) repeat_comparison`

# Student responses

---

1. `repeat_comparison <- births |> arrange(year, month, date_of_month) |> mutate( before = lag(births, 7), after = lead(births, 7), year2k = year - 2000 ) |> filter(month == date_of_month, month == year2k) |> select(year, month, date_of_month, before, births, after)`
2. `repeat_comparison <- births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7), yearm2k = year - 2000) |> filter(date_of_month == month, yearm2k == month) |> select(year, month, date_of_month, before, births, after)`
3. `repeat_comparison <- births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7), yearm2k = year-2000)|> filter(date_of_month == month, yearm2k == month) |> select(year, month, date_of_month, before, births, after) repeat_comparison`

# Student responses

---

1. `repeat_comparison <- births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7)) |> filter(month == date_of_month, date_of_month == year - 2000) |> select(year, month, date_of_month, before, births, after) repeat_comparison`
2. `repeat_comparison <- births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7)) |> filter(date_of_month == month, year - 2000 == month) |> select(year, month, date_of_month, before, births, after) repeat_comparison |> select(year, month, date_of_month, births, after)`
3. `repeat_comparison <- births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7)) |> filter(month == date_of_month, month == year - 2000) repeat_comparison |> select(year, month, date_of_month, before, births, after)`

# Student responses

---

1. `repeat_comparison <- births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7)) |> filter(date_of_month == month, month == (year - 2000)) repeat_comparison`
2. `repeat_comparison <- births |> arrange(year, month, date_of_month) |> mutate( before = lag(births, 7), after = lead(births, 7) ) |> filter(month == date_of_month & year - 2000 == month) repeat_comparison |> select(year, month, date_of_month, before, births, after)`
3. `repeat_comparison <- births |> arrange(year, month, date_of_month) |> mutate( before = lag(births, 7), after = lead(births, 7), yearm2k = year - 2000, ) |> filter(month == date_of_month, yearm2k == month) |> select(year, month, date_of_month, before, births, after)`



# Student responses

---

1. `repeat_comparison <- births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7), yearm2k = year - 2000) |> filter(date_of_month == month, month == year - 2000, date_of_month == year - 2000) |> select(year, month, date_of_month, before, births, after)`  
`repeat_comparison`
2. `repeat_comparison <- births |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7)) |> filter(date_of_month == month, month == year-2000) head(repeat_comparison)`
3. `repeat_comparison <- births |> arrange(year, month, date_of_month) |> mutate(before = lag(births,7), after = lead(births,7)) |> filter(month == date_of_month & month == (year - 2000) & date_of_month == (year - 2000))`
4. `repeat_comparison <- births_data %>% arrange(year, month, date_of_month) %>% mutate( before = lag(births, 7), after = lead(births, 7) ) %>% filter( month == date_of_month, month == year - 2000 ) %>% select(year, month, date_of_month, before, births, after)`

# Student responses

---

1. `repeat_comparison <- births_df |> arrange(year, month, date_of_month) |> mutate( before = lag(births, 7), after = lead(births, 7) ) |> filter(month == date_of_month, year - 2000 == month) |> select(year, month, date_of_month, before, births, after) repeat_comparison`
2. `repeat_comparison <- births|> arrange(year, month, date_of_month) |> mutate( before = lag(births, 7), after = lead(births, 7) ) |> filter( month == date_of_month, year - 2000 == month ) |> select(year, month, date_of_month, before, births, after) repeat_comparison`
3. `repeat_comparison <-births |> arrange(year,month,date_of_month)|> mutate( before=lag(births,7), after = lead(births,7), yearm2k = year -2000 )|> filter( date_of_month == month, yearm2k == month )`
4. `repeat_comparison |> arrange(year, month, date_of_month) |> mutate(before = lag(births, 7), after = lead(births, 7)) |> filter(month == date_of_month, date_of_month == (year - 2000)) |> select(year, month, date_of_month, before, births, after)`

- Keep `year` and the three counts: `before`, `births`, and `after`. - Gather the three count columns into `n\_births`, recording their names in `day`. - Keep the year attached to each count: three rows per year, 36 rows total. Save the table as `repeat\_long`.

---

72 anonymous responses

Other responses

72

# Student responses

---

1. `library(dplyr) library(tidyr) repeat_long <- repeat_comparison |> select(year, before, births, after) |> pivot_longer( cols = c(before, births, after), names_to = "day", values_to = "n_births" ) repeat_long`
2. `nlong -> repeat_comparison |> select(year, before, births, after) |> pivot_longer(cols = c(before, births, after), manes_to = "day", values_to = "n_births" )`
3. `repeat_comparison`
4. `repeat_comparison |> select(year, before, births, after) |> pivot_longer( cols = c(before, births, after), names_to = "day", values_to = "n_births" )`
5. `repeat_comparison |> select(year, before, births, after) |> pivot_longer( cols = c(before, births, after), names_to = "day", values_to = "n_births" )`
6. `repeat_comparison |> select(year, before, births, after) |> pivot_longer( cols = c(before, births, after) names_to = "day", values_to = "n_births" ) repeat_long`

# Student responses

---

1. `repeat_comparison |> select(year, before, births, after) |> pivot_longer( cols = c(before, births, after), names_to = "day", values_to = "n_births" )`
2. `repeat_comparison |> select(year, before, births, after) |> pivot_longer( cols = c(before, births, after), names_to = "day", values_to = "n_births" ) repeat_long`
3. `repeat_comparison |> select(year, before, births, after) |> pivot_longer( cols = c(before, births, after), names_to = "day", values_to = "n_births" )`
4. `repeat_comparison |> select(year, before, births, after) |> pivot_longer( cols = c(before, births, after), names_to = "day", values_to = "n_births")`
5. `repeat_comparison |> select(year, before, birth,after)|> pivot_longer( cols = c(before,birth,after), names_to = "day", values_to = "n-birth" )`
6. `repeat_comparison |> select(year, before, births, after)`
7. `repeat_comparison |> select(year, before, births, after) |> pivot_longer( cols = c(before, births, after), name_to = "day", values_to = "n_births")`

# Student responses

---

1. `repeat_comparison |> select(year, before, births, after) |> pivot_longer( cols = c(before, births, after), names_to = "day", values_to = "n_births" ) -> repeat_long`
2. `repeat_comparison |> select(year, before, births, after) |> pivot_longer( cols = c(before, births, after), names_to = "day", values_to = "n_births" )`
3. `repeat_comparison |> select(year, before, births, after) |> pivot_longer(cols = c(before, births, after), names_to = "day", values_to = "n_births")`
4. `repeat_comparison |> select(year, before, births, after) |> pivot_longer(cols = c(before, births, after), names_to = "day", values_to = "n_births")`
5. `repeat_comparison |> select(year, before, births, after) |> pivot_longer( cols = c(before, births, after), names_to = "day", values_to = "n_births" )`
6. `repeat_comparison|> select(year, before,births, after)|> pivot_longer( cols = c(before,births,after), names_to = "day", values_to = "n_births" )`

# Student responses

---

1. `repeat_comparison|> select(year, before, births, after)|> pivot_longer(cols = c(before, births, after), names_to = "day", values_to = "n_births")`
2. `repeat_long <- repeat_comparison |> select(year, before, births, after) |> pivot_longer( cols = c(before, births, after), names_to = "day", values_to = "n_births" )`
3. `repeat_long -> repeat_comparison |> select(year, before, births, after) |> pivot_longer(cols = c(before, births, after), name_to = "days", values_to = "n_births")`
4. `repeat_long <- repeat_comparison |> select(year, before, births, after) |> pivot_longer( cols= c(before, births, after), names_to = "day", values_to = "n_births" )`  
`repeat_long`
5. `repeat_long <- repeat %>% pivot_longer( cols = c(before, births, after), names_to = "day", values_to = "n_births" )`
6. `repeat_long <- repeat %>% pivot_longer( cols = c(before, births, after), names_to = "day", values_to = "n_births" )`

# Student responses

---

1. `repeat_long <- repeat |> pivot_longer( cols = c(before, births, after), names_to = "day", values_to = "n_births" )`
2. `repeat_long <- repeat_comparison |> pivot_longer( cols = c(before, births, after), names_to = "day", values_to = "n_births" )`
3. `repeat_long <- repeat_comparison |> select(year, before, births, after) pivot_longer( cols = c(before, births, after), names_to = "day", values_to = "n.birth" )`  
`repeat_long`
4. `repeat_long <- repeat_comparison |> select(year, before, births, after) |> pivot_longer( cols = c(before, births, after), names_to = "day", values_to = "n_births" )`
5. `repeat_long <- repeat_comparison |> select(year, before, births, after) |> pivot_longer( cols = c(before, births, after), names_to = "day", values_to = "n_births" )`
6. `repeat_long <- repeat_comparison |> select(year, before, births, after) |> pivot_longer( cols = c(before, births, after), names_to = "day", values_to = "n_births" )`



# Student responses

---

1. `repeat_long <- repeat_comparison |> select(year, before, births, after) |> pivot_longer( cols = c(before, births, after), names_to = "day", values_to = "n_births" )`
2. `repeat_long <- repeat_comparison |> select(year, before, births, after) |> pivot_longer( cols = c(before, births, after), names_to = "day", values_to = "n_births" )`
3. `repeat_long <- repeat_comparison |> select(year, before, births, after) |> pivot_longer( cols = c(before, births, after), names_to = "day", values_to = "n_births" )`  
`repeat_long`
4. `repeat_long <- repeat_comparison |> select(year, before, births, after) |> pivot_longer( cols = c(before, births, after), names_to = "day", values_to = "n_births" )`  
`repeat_long`
5. `repeat_long <- repeat_comparison |> select(year, before, births, after) |> pivot_longer( cols = c(before, births, after), names_to = "day", values_to = "n_births" )`  
`repeat_long`

# Student responses

---

1. `repeat_long <- repeat_comparison |> select(year, before, births, after) |> pivot_longer( cols = c(before, births, after), names_to = "day", values_to = "n_births")`
2. `repeat_long <- repeat_comparison |> select(year, before, births, after) |> pivot_longer( cols = c(before, births, after), names_to = "day", values_to = "n_births" )`  
`repeat_long`
3. `repeat_long <- repeat_comparison |> select(year, before, births, after) |> pivot_longer(cols = c(before, births, after), names_to = "day", values_to = "n_births")`  
`repeat_long`
4. `repeat_long <- repeat_comparison |> select(year, before, births, after) |> pivot_longer(cols = c(before, births, after), names_to = "day", values_to = "n_births" )`
5. `repeat_long <- repeat_comparison |> select(year, before, births, after) |> pivot_longer(cols = c(before, births, after), names_to = "day", values_to = "n_births")`

# Student responses

---

1. `repeat_long <- repeat_comparison |> select(year, before, births, after) |> pivot_longer( cols = c(before, births, after), names_to = "day", values_to = "n_births")`  
`repeat_long`
2. `repeat_long <- repeat_comparison |> select(year, before, births, after) |> pivot_longer(cols = c(before, births, after), names_to = "day", values_to = "n_births")`  
`repeat_long`
3. `repeat_long <- repeat_comparison |> select(year, before, after, births) |> pivot_longer( cols = c(before, births, after), names_to = "day", values_to = "n_births")`
4. `repeat_long <- repeat_comparison |> select(year, before, births, after) |> pivot_longer( cols = c(before, births, after), names_to = "day", values_to = "n_births" )`
5. `repeat_long <- repeat_comparison |> select(year, before, births, after) |> pivot_longer( cols = c(before, births, after), names_to = "day", values_to = "n_births" )`
6. `repeat_long <- repeat_comparison |> select(year, before, births, after) |> pivot_longer( cols = c(before, births, after), names_to = "day", values_to = "n_births")`

# Student responses

---

1. `repeat_long <- repeat_comparison |> select(year, before, births, after) |> pivot_longer( cols = c(before, births, after), names_to = "day", values_to = "n_births" )`
2. `repeat_long <- repeat_comparison |> select(year, before, births, after) |> pivot_longer(cols = c(before, births, after) names_to = "day", values_to = "n_births" )`
3. `repeat_long <- repeat_comparison |> select(year, before, births, after) |> pivot_longer(cols = c(before, births, after), names_to = "day" values_to = "n_births")`  
`|>`
4. `repeat_long <- repeat_comparison |> select(year, before, births, after) |> pivot_longer(cols = c(before, births, after), names_to = "day", values_to = "n_births")`
5. `repeat_long <- repeat_comparison |> select(year, before, births, after) |> pivot_longer(cols = c(before, births, after), names_to = "day", values_to = "n_births")`  
`repeat_long`

# Student responses

---

1. `repeat_long <- repeat_comparison |> select(year, before, births, after) |> pivot_longer(cols = c(before, births, after), names_to = "day", values_to = "n_births")`
2. `repeat_long <- repeat_comparison |> select(year, before, births, after) |> pivot_longer(cols = c(before, births, after), names_to = "days", values_to = "n_births")`
3. `repeat_long <- repeat_comparison |> select(year, before, births, after) |> pivot_longer(cols = c(before, births, after), names_to = "day", values_to = "n_births")`
4. `repeat_long <- repeat_comparison |> select(year, before, births, after) |> pivot_longer(cols = c(before, births, after), names_to = "day", values_to = "n_births")`  
`print(repeat_long)`
5. `repeat_long <- repeat_comparison |> select(year, before, births, after) |> pivot_longer(cols=c(before, births, after), names_to="day", values_to="n_briths")`
6. `repeat_long <- repeat_comparison |> pivot_longer( cols = c(before, births, after), names_to = "day", values_to = "n_births")`

# Student responses

---

1. `repeat_long <- repeat_comparison |> select(year, before, births, after) |> pivot_longer( cols = c(before, births, after), names_to = "day", values_to = "n_births" )  
repeat_long`
2. `repeat_long <- repeat_comparison |> select(year, before, births, after) |> pivot_longer( cols = c(before, births, after), names_to = "day", values_to = "n_births" )`
3. `repeat_long <- repeat_comparison |> select(year, before, births, after) |> pivot_longer ( cols = c(before, births, after), names_to = "day", values_to = "n_births")  
|> print()`
4. `repeat_long <- repeat_comparison |> select(year, before, births, after) |> pivot_longer(cols = c(before, births, after), names_to = "day", values_to = "n_births" )`
5. `repeat_long <- repeat_comparison |> select(year, before, births, after) |> pivot_longer(cols = c(before, births, after), names_to = "day", values_to "n_births")`
6. `repeat_long <- repeat_comparison |> select(year, before, births, after) |> pivot_longer(cols=c(before, births, after), names_to = 'day', values_to = 'n_births')`

# Student responses

---

1. `repeat_long <- repeat_comparison|> select(year, before, births, after)|> pivot_longer( cols = c(before, births, after), names_to = "day", values_to = "n_births" )  
repeat_long`
2. `repeat_long <-repeat_comparison |> select(year, before, births, after)|> pivot_longer(cols=c(before, births, after), names_to = "day", values_to = "n_births")  
repeat_long`
3. `repeat_long <-repeat_comparison |> select(year, before, births, after) |> pivot_longer( cols = c(before, births, after), names_to = "day", values_to = "n_births" )  
repeat_long`
4. `repeat_long <-repeat_comparison|> select(year,before,births,after)|> pivot_longer(cols = c(before, births, after), names_to = "day", values_to = "n_births")`
5. `repeat_long<- repeat_comparison|> select(year,before,births,after)|> pivot_longer(cols=c(before,births,after), names_to="day", values_to="n_births")`
6. `repeat_long<-repeat_comparison |> select(year, before, births, after) |> pivot_longer( cols = c(before, births, after), names_to = "day", values_to= "n_births" )`

# Student responses

---

1. `repeat_long<-repeat_comparison|> select(year, before,births, after)|> pivot_longer(cols = c(before, births, after), names_to = "day", values_to = "n_births")`  
`repeat_long`
2. `repeat_longer = repeat_comparison |> select(year, before, births, after) |> pivot_longer( cols = c(before, births, after), names_to = "day", values_to = "n_births"`  
`) repeat_longer`



**- Map ``day`` to x, ``n_births`` to y, and ``day`` to fill. - Add bars with ``geom_col()``. - Add ``facet_wrap(~year)`` to give each year its own panel.**

---

66 anonymous responses

Other responses

66

# Student responses

---

1. `births |> ggplot(aes(x = day, y = n_births, fill = day)) + geom_col() + facet_wrap(~ year)`
2. `library(dplyr) library(ggplot2) repeat_long |> mutate(day = factor( day, levels = c("before", "births", "after") )) |> ggplot(aes(x = day, y = n_births, fill = day)) + geom_col() + facet_wrap(~ year) + labs( x = "Day relative to repeating date", y = "Number of births", fill = "Day" ) + theme_minimal()`
3. `repeat_long |> ggplot() + geom_col(aes(x = day, y = n_births, fill = day), position = "dodge") + facet_wrap(~ year)`
4. `repeat_long |> ggplot() + geom_col(aes(x = year, y = n_births, fill = day), position = "dodge") + facet_wrap(~ year)`
5. `repeat_long |> ggplot() + geom_col(aes(x= year, y = n_births, fill = day) position = "dodge") + facet_wrap (year)`
6. `repeat_long |> ggplot() + geom_col(aes(x = day, y = n_births, fill = day),+ positon = "dodge") + facet_wrap(~year)`
7. `repeat_long |> ggplot() + geom_col(aes(x = year, y = n_births, fill = day))`

# Student responses

---

1. `repeat_long |> ggplot() + geom_col(aes(x = day, y = n_births, fill = day), position = "dodge") + facet_wrap(~ year)`
2. `repeat_long |> ggplot() + geom_col(aes(x = day, y = n_births, fill = day), position = "dodge") + facet_wrap(~ year)`
3. `repeat_long |> ggplot() + geom_col(aes(x = year, y = n_births, fill = day), position = "dodge") + facet_wrap(~ year)`
4. `repeat_long |> ggplot() + geom_col(aes(x = year, y = n_births, fill = day), position = "dodge") + facet_wrap(~year)`
5. `repeat_long |> ggplot() + geom_col(aes(x = year, y = n_births, fill = day), position = "dogge")`
6. `repeat_long |> ggplot() + geom_col(aes(x = year, y = n_births, fill = day), position = "dodge") + facet_`
7. `repeat_long |> ggplot() + geom_col (aes(x= year, y = n_births, fill = day), position "dodge") + facet_wrap(~year)`
8. `repeat_long |> ggplot() + geom_col(aes(x = year, y = n_births, fill = day), position = "dodge") + facet_wrap(~ year)`
9. `repeat_long |> ggplot() + geom_col(aes(x=year, y=n_births, fill = day), position = "dodge") + facet_wrap(~year)`

# Student responses

---

1. `repeat_long |> ggplot() + geom_col(aes(x = day, y = n_births, fill = day), position = "dodge") + facet_wrap(~ year)`
2. `repeat_long |> ggplot() + geom_col(aes(x = year, y = n_births, fill = day), position = "dodge")`
3. `repeat_long |> ggplot() + geom_col(aes(x = year, y = n_births, fill = day), position = "dodge") + facet_wrap(~year)`
4. `repeat_long |> ggplot() + geom_col(aes(x = year, y = n_births, fill = day), posiiton = "dodge")`
5. `repeat_long |> ggplot() + geom_col(aes(x = year, y = n_births, fill = day), position = "dodge") + facet_wrap(~year)`
6. `repeat_long |> ggplot() + geom_col(aes(x = year, y = n_births, fill = day), position = "dodge") + facet_wrap(~ year)`
7. `repeat_long |> ggplot() + geom_col(aes(x = year, y = n_births, fill = day))`
8. `repeat_long |> ggplot() + geom_col(aes(x = year, y= n_births, fill = day), position = "dodge") + facet_wrap(~year)`
9. `repeat_long |> ggplot() + geom_col(aes(x = year, y=n_births, fill = day), position = "dodge") + facet_wrap(~4)`

# Student responses

---

1. repeat\_long |> ggplot() + geom\_col(aes(x = year, y=n\_births, fill = day), position="dodge") + facet\_wrap(~ year)
2. repeat\_long |> ggplot() + geom\_col(aes(x= year, y = n\_births, fill = day, position = "dodge")) + facet\_wrap(~ year)
3. repeat\_long |> ggplot() + geom\_col(aes(x= year, y= n\_births, fill = day), position = "dodge") + facet\_wrap(~year)
4. repeat\_long |> ggplot() + geom\_col(aes(x=year, y=n\_births, fill=day), position = "dodge") + facet\_wrap(~ year)
5. repeat\_long |> ggplot() + geom\_col(aes(x=year, y=n\_births, fill=day), position="dodge")
6. repeat\_long |> ggplot() + geom\_col(aes(x = year, y = n\_births, fill = day), position = "dodge")+ facet\_wrap(~year)
7. repeat\_long |> ggplot()+geom\_col(aes(x=day,y=n\_births, fill=day), position="dodge")+ facet\_wrap(~year)
8. repeat\_long |> ggplot(aes(x = day, y = n\_births, fill = day)) + geom\_col() + facet\_wrap(~ year)
9. repeat\_long |> ggplot(birth\_counts, aes(x = day, y = n\_births, fill = day)) + geom\_col() + facet\_wrap(~ year)

# Student responses

---

1. `repeat_long |> ggplot(births_by_day_year, aes(x = day, y = n_births, fill = day)) + geom_col() + facet_wrap(~ year)`
2. `repeat_long |> mutate( day = factor( day, levels = c("before", "births", "after") ) ) |> ggplot(aes(x = day, y = n_births, fill = day)) + geom_col() + facet_wrap(~ year) + labs( title = "Birth Counts Before, On, and After the Repeating Date", x = "Day", y = "Number of Births" )`
3. `repeat_long |> mutate(day = factor( day, levels = c("before", "births", "after") ) ) |> ggplot(aes(x = day, y = n_births, fill = day)) + geom_col() + facet_wrap(~ year)`
4. `repeat_long |> mutate(day = factor( day, levels = c("before", "births", "after") ) ) |> ggplot(aes(x = day, y = n_births, fill = day)) + geom_col() + facet_wrap(~ year) + labs( title = "Births on Memorable Dates Compared to Nearby Dates", x = "Day", y = "Number of Births" )`
5. `repeat_long |> mutate(day = factor( day, levels = c("before", "births", "after")))` `ggplot()`

# Student responses

---

1. `repeat_long |> mutate(day = factor( day, levels = c("before", "births", "after")))) ggplot()`
2. `repeat_long |> mutate(day = factor( day, levels = c("before", "births", "after")))) ggplot() + geom_col(aes(x = year, y = n_births, fill = day, position = "dodge"))`
3. `repeat_long |> mutate(day = factor( day, levels = c("before", "births", "after")))) ggplot() + geom_col(x = year)`
4. `repeat_long |> mutate(day = factor( day, levels = c("before", "births", "after")))) ggplot() + geom_col(aes(x = year, y = n_births, fill = day))`
5. `repeat_long |> mutate(day = factor( day, levels = c("before", "births", "after")))) ggplot() + geom_col(aes(x = year, y = n_births, fill = day))`
6. `repeat_long |> mutate(day = factor( day, levels = c("before", "births", "after")))) ggplot(aes(x = day, y = n_births, fill = day)) + geom_col() + facet_wrap(~ year))`

# Student responses

---

1. `repeat_long |> mutate(day = factor( day, levels = c("before", "births", "after")))|> ggplot() + geom_col(aes(x = day, y = n_births, fill = day), position = "dodge") + facet_wrap(~ year)`
2. `repeat_long |> mutate(day = factor( day, levels = c("before", "births", "after")))|> ggplot() + geom_col(aes(x = day, y = n_births, fill = day)) + facet_wrap(~ year)`
3. `repeat_long |> mutate(day = factor( day, levels = c("before", "births", "after")))|> ggplot() + geom_col(aes(x = year, y = n_births, fill = day), position = "dodge") + facet_wrap(~year)`
4. `repeat_long |> mutate(day = factor( day, levels = c("before", "births", "after")))|> ggplot() + geom_col(aes(x = year, y = n_births, fill = day))`
5. `repeat_long |> mutate(day = factor( day, levels = c("before", "births", "after")))|> ggplot() + geom_col(aes(x = year, y = n_births,fill = day)) + facet_wrap(~ year)`
6. `repeat_long |> mutate(day = factor( day, levels = c("before", "births", "after")))|> ggplot() + geom_col(aes(x= year, y = n_births, fill = day))`



# Student responses

---

1. `repeat_long |> mutate(day = factor( day, levels = c("before", "births", "after")))|> ggplot() + geom_col(aes(x=year, y=n_births, fill = day))`
2. `repeat_long |> mutate(day = factor( day, levels = c("before", "births", "after")))|> ggplot(births_counts, aes(x = day, y = n_births, fill = day)) + geom_col() + facet_wrap(~ year)`
3. `repeat_long |> mutate(day = factor(day, levels = c("before", "births", "after")))|> ggplot(repeat_long, aes(x = factor(day), y = n_births, fill = factor(day))) + geom_col() + facet_wrap(~ year)`
4. `repeat_long |> mutate(day = factor(day, levels = c("before", "births", "after")))|> ggplot(aes(x = day, y = n_births, fill = day)) + geom_col() + facet_wrap(~ year)`
5. `repeat_long |> mutate(day = factor(day, levels = c("before", "births", "after")))|> ggplot(aes(x = day, y = n_births, fill = day)) + geom_col() + facet_wrap(~ year)`
6. `repeat_long |> mutate(day = factor(day, levels = c("before", "births", "after")))|> ggplot(aes(x = day, y = n_births, fill = day)) + geom_col() + facet_wrap(~ year)`

# Student responses

---

1. `repeat_long |> mutate(day = factor(day, levels = c("before", "births", "after")))|> ggplot(aes(x = day, y = n_births, fill = day)) + geom_col() + facet_wrap(~ year) + labs(x = NULL, y = "Births", fill = NULL)`
2. `repeat_long |> ggplot() + geom_col(aes(x = year, y=n_births, fill = day), position = "dodge") facet_wrap(~year)`
3. `repeat_long |> ggplot() + geom_col(aes( x = year, y = n_births, fill = day), position = "dodge") + facet_wrap(~ year)`
4. `repeat_long |> gg_plot()+ geom_col(aes(x=year, y=n_births, fill=day), position="dodge")`
5. `repeat_long |> ggplot() + geom_col(aes(x=year, y = n_births, fill = day), position = "dodge") + facet_wrap(~year)`
6. `repeat_long |> ggplot() + geom_col(aes(x = year, y = n_births, fill = day) position = "dodge") + # mutate(day = factor( # day, # levels = c("before", "births", "after")))) # ggplot()`
7. `repeat_long |> ggplot(aes(x = day, y = n_births, fill = day)) + geom_col() + facet_wrap(~ year)`
8. `repeat_long |> ggplot() + geom_col(aes(x=day, y=n_births, fill = day, position = 'dodge')) + facet_wrap(~ year)`

# Student responses

---

1. `repeat_long |> ggplot() + geom_col(aes(x = year, y = n_births, fill = day), position = "dodge") + facet_wrap(~year)`

# What do these plots tell us? What patterns did you find? Did you notice any exceptions or unusual dates?

---

78 anonymous responses

## Vague or placeholder remark

20

Noninformative or incomplete response that does not describe patterns or specifics.

## Seasonality and Repeat prominence

14

Student noted that the Repeat group is prominent and mentioned seasonal patterns (response contains typos but conveys seasonality and popularity of Repeat data).

## Recent increase observed

11

Student observed that the measured quantity is higher in recent time periods.

## No clear pattern observed

9

Student reported that they did not observe any discernible pattern in the plots.

# Response themes for question 15

|                                                                                                                            |   |
|----------------------------------------------------------------------------------------------------------------------------|---|
| <b>Suggests certain birthdays are more sought</b>                                                                          | 7 |
| Unclear or typo-filled comment implying that some birthdays are more sought after or popular than others.                  |   |
| <b>Higher average for Repeat category</b>                                                                                  | 6 |
| Student identified that the 'Repeat' group shows a statistically higher average number of births compared with others.     |   |
| <b>No consistent increase on special dates</b>                                                                             | 6 |
| Student observed that births do not consistently rise on special or 'notable' dates.                                       |   |
| <b>Year-to-year variation or specific exceptions</b>                                                                       | 5 |
| Student pointed out that some years differ (middle years peak) or mentioned particular years with increases as exceptions. |   |

# Student responses

---

1. 2006 seems to have relatively low numbers on 06/06/2006 compared to the weeks before and after. This is likely due to the superstition around the number 666.
2. for most part, the special date is more common than not; but some years there were less 05, 10
3. generally more people was born on year, month, and date that are the same but there are exceptions
4. Honestly, its a very slight increase for most times. Generally its just a slight bump between the lower and higher dates. 2001 and 2004 had some exceptions but really everything else shows a slight bump
5. I think the plots show that generally there are more births on the repeated days, however it doesn't seem like the difference is significant and it isn't true for all repeated birthdays
6. In general, there seems to be a slight increase on births for the unusual days, but there are some exceptions, like 1/1/2001.
7. In most years there are slightly higher birth rates on these dates, however in 2001 the week after there was a significant increase in births

# Student responses

---

1. It is generally true that people seek out weird birthdays, but this is not the case on 1/1/1 because that is new year's day and so this could be the plausible cause of there being less births
2. it seems like most years the days that are sought out for births have a higher birth rate, however exceptions could be around holidays and other major days that hurt scheduling
3. Most repeated dates have more births than the same weekday one week before or after, especially 08/08/08, 09/09/09, 11/11/11, and 12/12/12.
4. Some of the years show a slight increase in births on the particular date, but its not consistent.
5. The plots show that most memorable dates have more births than the surrounding dates, suggesting that some parents may schedule deliveries on these days. However, 2001, 2004, and 2006 are exceptions, with fewer births on the repeated dates compared to at least one nearby date.
6. The plots tell us that people generally try to go for- or avoid- these certain threepeat dates, although some do not seem to have significance.
7. There are more repeated dates in sien years

# Student responses

---

1. There are some very interesting patterns in the data. We see that only in certain times do people aim for the more memorable days. The exceptions to this are 1/1/2001 and 6/6/2006 which is not all that surprising because these days/numbers have other associations too
2. There are some years where people seek out these special birthdays, but there are some exceptions in some of those years
3. There is not necessarily a fixed pattern based on these plots.
4. These plots show that over many of the past years, there has been an increase in the number of births on days with the same day, month, and year value, however it is not consistent across every year.
5. These plots tell us that some years have more people born on the special days compared to the before and after days, while other years are less conclusive. One notable exception is 2001, which makes sense, as that is new years day and people are unlikely to schedule c-sections for them



# Student responses

---

1. these plots tell us there are lower births on weekends and this is true even on holidays, where they are already low even if they are on weekdays, and on weekends, they are even lower.

# Student responses

---

1. 6 6 6 is notadable lower which i think is fucnny but teh day of the week is still totally controled and not every reperting number shows and increased births
2. Almost all of the plots had data that said the births were more common than the before and after dates but it was only slightly higher, some of them were still lower than before and after
3. At 2001 the special day's birth is lower than normal
4. I don't see very much of a difference, I don't think that there is a pattern
5. I think the weird birthdays became more popular as time went on because the difference between the weird and normal birthdays appeared to increase. Its possible that some of these days are coinciding with other eventful days.
6. In certain years like 2008, 2003, and 2009, repeated dates are more frequent. However, births also seem more frequent in general in particular years around these repeated dates.
7. Most of the avg births for the memorable dates are higher except for 2001 and 2006.

# Student responses

---

1. Most repeated dates had more births than the weeks before and after. This may be because people choose memorable dates for scheduled births. However, 2001 and 2006 were exceptions.
2. nr
3. There aren't any exceptions that stand out
4. There is a small increase in repeating. But it was not consistent across all of the graphs. Statistical testing is needed.
5. These plots tell us that over time, the day that was repeated generally had more births compared to 7 days before and 7 days after. In 2001, the 7 days after bar had more than the repeated date and the 7 days before bar.
6. We are able to see that the repeated date is very popular and there is seasonality involved and there are also patterns we see about the weekend and weekdays
7. We see that there are multiple repeated date patterns are found.

# Student responses

---

1. Birth counts vary systematically across the calendar. For some years, memorable dates differ from nearby dates, but the overall pattern is not clear.
2. its higher recently
3. more births in the 2nd half of the selection of years, less births on holidays and more on the weekdays
4. No, but 2008 and 2009 seem to have a lot of births.
5. Number of births varies each year
6. people started seeking out these birthdays after 2006. because they realized they were special after that year
7. The plot show that births for the most part have increased in more recent years. there was an exception of 2004 and 2010 where it was lower than expected
8. There are a lot more repeated dates in 2009, 2008, 2011
9. There are some increases, but there are even some decreases in the number of births. It seems like the hypothesis generally holds.

# Student responses

---

1. There seems to be a slight increase in these repeated date birthdays in each year after 2006, but it doesn't look like a huge difference.
2. you cant accuratly hit the date, you can get within the range of it though

# Student responses

---

1. 2004 and 2010 has less births than others
2. In general we see a lot of repeated birthdays that are higher than the week before and the week after, with 2010 being an exception.
3. The unusual days tend to have more births than a week before and after.
4. There appears to be no pattern
5. There doesn't appear to be any patterns.
6. These plots tell us that in general, people do tend to seek out these weird birth dates more frequently than the weeks before or after. There was one clear exception, which was 01/01/01, which showed an unusual lack of births compared to the rest of the dataset, but can be explained because it is a holiday.
7. they are the same
8. unimodal
9. weird patterns. equal distribution

# Student responses

---

1. 2008, 2009, 2012 has more people born on these special days compared to a week before or after compared to other years.
2. For some of the years, the repeated birth date is higher than the before and after but not for 2001. This could be because it was new years
3. Generally there is a tendency that the special days have greater birth than other days. But some exceptions still exist like 2001
4. Generally true that people seek out the odd birthdays but there are exceptions. They are only slightly higher than before and after in both.
5. It makes sense 6/6/6 has less birthdays. Generally, the other days, months, years that match have more birthdays.
6. These plots tell us generally that the special birthdays have more births but are pretty much uniformly distributed with the special birthdays only being marginally higher
7. These plots tell us if these birthdays are more frequent than similar days surrounding it. I didn't really notice any exceptions.

# Student responses

---

1. it looks like there is slightly more births on repeated dates but not by a super large amount, this could jsut be due to chance
2. On average, the repeated day is slightly higher than the others but that is not the case for all of them. Weekends and holidays surrounding the days could be a factor.
3. Repeat seems to have a statistically higher average number of births.
4. The year 2010 seemed to have significantly less births than the other yeas. Also, repeat was the most common category among mostly all the years
5. There are more people looking for birthdays with the month, day and year all the same cause the middle bar is the highest.
6. There is more people seeking out birthdays with the month, day, and year all the same because the middle bar is the highest one.



# Student responses

---

1. I think besides 2005, the before and after bars were equal and then the births bar was either above or below the before and after bars.
2. I think I would think that there are more people seeking out these bdays then there are
3. I think in more recent years people were more likely to seek out a memorable birthday. 2006 seems to be an outlier since 666 is considered an unlucky number among Christians, which is the dominant religion in the US.
4. most of the dates had higher birth rates except for 6/6/6.
5. The plots tell us that births do not consistently increase on "special" birthdays
6. There are slightly more people seeking out these birthdays. 01/01/01 is an exception because its new years. 06/06/06 is also an exception because its the devil number

# Student responses

---

1. 2001 and 2004 has fewer births
2. In most years, the same dates do have more births than the previous and after week. The big exception is 1/1/2001, I hypothesize this is because of new years and the new century so people didn't want to miss celebrations
3. middle of the year always more
4. Most of these days have more births on the day, but in 2001 and 2006 there were exceptions, where there was more births before or after 7 days. For 1-1-01, probably because of Christmas and 6-6-06
5. Only some years were the middle the highest. Some increased like 2001. They were controlled to be all the same day of the week. I guess 2001 was a holiday

- Starting with all rows of ``vietnam_casualties``, count the number of people at each ``age``. - Map ``age`` to x and your count column to y. - Use ``geom_col()`` to draw one bar per age. - Label the axes and title the plot to identify U.S. military deaths in the Vietnam War.

---

46 anonymous responses

Other responses

46

# Student responses

---

1. `library(dplyr) library(ggplot2) vietnam_casualties |> count(age) |> ggplot(aes(x = age, y = n)) + geom_col(fill = "steelblue") + labs( title = "U.S. military deaths in the Vietnam War by age", x = "Age (years)", y = "Number of deaths" ) + theme_minimal()`
2. `vietnam_casualties`
3. `vietnam_casualties`
4. `vietnam_casualties`
5. `vietnam_casualties`
6. `vietnam_casualties`
7. `vietnam_casualties`
8. `vietnam_casualties`
9. `vietnam_casualties`

# Student responses

---

1. vietnam\_casualties
2. vietnam\_casualties
3. vietnam\_casualties
4. vietnam\_casualties
5. vietnam\_casualties %>% count(age) %>% ggplot(aes(x = age, y = n)) + geom\_col() + labs( title = "U.S. military deaths in the Vietnam War", x = "Age", y = "Number of deaths" )
6. vietnam\_casualties |>
7. vietnam\_casualties |> count(age) |> ggplot() + geom\_col( aes(x = age, y = count) )
8. vietnam\_casualties |> count(age) |> ggplot(aes(x = age, y = n)) + geom\_col() + labs( x = "Age", y = "Number of Deaths", title = "U.S. Military Deaths in the Vietnam War by Age" )
9. vietnam\_casualties |> ggplot() + geom\_bar(aes(x = age))

# Student responses

---

1. `vietnam_casualties |> ggplot() + geom_bar(aes(x = age))`
2. `vietnam_casualties |> count(age)`
3. `vietnam_casualties |> count(age) |> ggplot() + geom_bar(aes(x = age, y = count(age))) + labs (title = "Casualties in Vietnam War due to Age", x = "Age", y = "Frequency")`
4. `vietnam_casualties |> count(age) |> ggplot(aes(x=age, y = n)) + geom_col()`
5. `vietnam_casualties |> ggplot() + geom_bar(aes(x = age))`
6. `vietnam_casualties |> ggplot() + geom_bar(aes(x = age))`
7. `vietnam_casualties |> ggplot() + geom_bar(aes(x=age))`
8. `vietnam_casualties |> ggplot() + geom_bar(aes(x=age))`
9. `vietnam_casualties |> ggplot() + geom_bar(aes(x=age))`

# Student responses

---

1. `vietnam_casualties |> ggplot() + geom_col(aes(x = age))`
2. `vietnam_casualties |> group_by(age) |> count(age)`
3. `vietnam_casualties |> ggplot + geom_bar(aes(x=age))`
4. `vietnam_casualties |> count(age)`
5. `vietnam_casualties |> filter(name = "Davis")`
6. `vietnam_casualties |> ggplot() + geom_bar(aes(x = age))`
7. `vietnam_casualties |> ggplot() + geom_bar(aes(x = age))`
8. `vietnam_casualties |> ggplot() + geom_bar(aes(x = age))`
9. `vietnam_casualties |> ggplot() + geom_bar(aes(x = age))`

# Student responses

---

1. `vietnam_casualties |> ggplot() + geom_bar(aes(x = age))`
2. `vietnam_casualties |> ggplot() + geom_bar(aes(x = age))`
3. `vietnam_casualties |> ggplot() + geom_bar(aes(x = age))`
4. `vietnam_casualties |> ggplot() + geom_bar(aes(x=age))`
5. `vietnam_casualties |> ggplot() + geom_bar(aes(x=age))`
6. `vietnam_casualties |> ggplot() +geom_bar(aes(x = age))`
7. `vietnam_casualties |> group_by(age)|> count(age)`
8. `vietnam_casualties |> mean(age) |> print()`
9. `vietnam_casualties|> group_by(age)|> count(age)`



# Student responses

---

1. vietnam\_casualties|> group\_by(age)|> summarize(n=n())|>

# In your opinion, what does it mean for the draft to be fair?

---

81 anonymous responses

## Equal probability for every day

38

Fairness means each calendar day has the same chance of being selected; no days favored or disadvantaged.

## Equal chance for each eligible person

19

Fairness means every eligible individual has the same opportunity to be selected, independent of personal attributes like birthday or background.

## No pattern or bias in selection

7

Fairness means selections show no detectable pattern or systematic bias in which days are picked.

## Unclear or negative response

6

Responses that are brief, unclear, or express disagreement without a clear alternative definition of fairness.

# Response themes for question 17

---

## Fairness as evenness across groups

4

Fairness means data processing or filtering should not advantage particular groups; outcomes should not benefit certain groups over others.

---

## Other responses

4

## Completely random selection of dates

3

Fairness means dates are drawn via a fully random process with unpredictable outcomes.

---

# Student responses

---

1. A real random select
2. all dates have equal likelihood to be pulled.
3. All of the dates have an equal chance of being chosen
4. Any birth was equally as likely to be drawn as any other date of the year.
5. each birth day have the same probabiltiy to be chosen
6. Each day has the equal chance to be selected.
7. equal chance for every date to be chosen.
8. equal chance for every date to be drawn
9. Equal likelihood of getting drafted

# Student responses

---

1. every date having the exact same chance of being selected first and then continue to be selective at. anormal rate without any bais in months
2. every date must be in the urn and only once. when they choose the card, they should choose randomly without bias
3. Every date should be equally likely to be chosen
4. Every single day would have the same probability of being drawn at any given point
5. Fair means that every day has an equal chance of being picked.
6. for each date to have an equally likely chance to be drawn. look at what months had the most days drawn
7. for the draft to be fair it would mean that every birthday has the same chance of being pulled
8. For the draft to be fair there would have to be an equal likelihood that every day of the year could be chosen. However, even this is still not really fair.
9. For the draft to be fair, each birthday had to have an equal chance of being picked. To measure this, we need to see if one month or one day had a higher likelihood of being chosen.

# Student responses

---

1. For the draft to be fair, each birthday odds of getting picked would be even
2. for the draft to be fair, every birthday should have the same probability of being drawn in any order
3. For the draft to be fair, the odds of any one birthday being drawn should be the same as any other day being drawn
4. For the draft to be fair, the values for each date would need to be evenly mixed, with no extra quantities or missing quantities for certain dates. They would also have to be the same size and shape for selection.
5. For the draft to be fair, there must be equal odds for all 366 days to be drawn. It must be completely random, and without any kind of layering order or human involvement- as was done in the vietnam draft.
6. give every birthday an equal chance of receiving an early or late lottery number
7. I believe fairness in this case means all the birthdays are equally likely to be selected
8. I think it has to be 100% random for a draft to be fair
9. I think it would be fair if every single day had an equal chance of being chosen

# Student responses

---

1. If the draft was fair, there would be an equal probability for every day to get drafted
2. It is unfair anyway but it is even selection or an equal chance at being chosen for it.
3. It means that every birthday has an equal chance of getting picked.
4. It means that every birthday has the same probability of being selected for the draft
5. it means to be completely random and with even likelihood of any outcome.
6. it was meant to random in selection
7. My assumption is that every person has an equal chance of being selected, so every birth date has an equal chance of being selected would be fair.
8. That each birthday has an equally likely chance of being selected, such that there is no observable trend to the picked birthdays.
9. The birthdates of each month were evenly distributed across the draft

# Student responses

---

1. There should be an equal chance of selection, and here we need to ensure that the probability of being selected is equal, regardless of circumstance. For example, paper slips at the bottom of the jar might have been less likely to be chosen.
2. To me, for the draft to be fair, I think they would have needed to put every possible date into a container, and then shaken the container up to randomize the order and the layers of the birthdays.



# Student responses

---

1. a fair draft means everyone who is eligible has an equal chance of being selected.
2. A fair draft would give every eligible person an equal chance of being selected, regardless of their birthday, wealth, or background. The selection process should be random and free from bias.
3. air if every eligible person has an equal chance of being selected
4. equal chance for every person to be picked
5. every person has an equal chance of getting drafted
6. everyone has an equal chance of being selected
7. Everyone should have an equal chance of being pulled. Birthday was probably the most fair way but there is skew with dates
8. Fairness here assumes you're equally likely as all other draftees to be drafted.
9. Fairness means each person has an equal likelihood of being drafted, not each birthday has an equal likelihood because some birthdays are more common than others.

# Student responses

---

1. For all eligible people to have an equal chance
2. For the draft to be fair would mean every person(in a certain age range) would have an equal chance to be selected.
3. For the draft to be fair, all eligible individuals should have an equal chance of being called early. However, because birthdays were drawn randomly, and birth dates are NOT a uniform distribution, eligible individuals likely do not have an equal chance of being drafted early.
4. I would say the draft to be fair is that each person has an equal chance to be picked or not picked for draft and there is not a larger competitive edge. In the case of the draft I would say no as there are more people born on certain days than others, so you could potentially pull way more than another day and would be unfair for grouping
5. It means that every person has an equal chance of being chosen, but also accounts for those who may not be phsycially able to participate or other circumstances
6. That everyone who is available should have equal chance to be drafted.

# Student responses

---

1. The draft should be fair so that everyone is given an even opportunity to become a soldier.
2. The draft would be fair if everyone that was eligible for the draft had the same number of entries in the drawing so that every man that can be drafted has the exact same likelihood of getting drafted as any other man
3. The draft would be fair if the birthdays were truly randomly selected. Honestly, I think it would be more fair if everyone was assigned a number and these numbers were individually drawn because certain birthdays have higher quantities of people.
4. To me, a fair draft would mean each person eligible is equally likely to be drafted into the Vietnam war.

# Student responses

---

1. A fair draft would have no patterns and be completely random
2. Days are actually randomly chosen. Can measure the most common birthdays and their order in the draft.
3. I think fairness means how random the selections are. If its not random there would be more birthdays around a specific day or some sort of bias going on.
4. It means that distribution of birthdays selected does not follow an abnormal skew.
5. no discernable pattern for the selected days
6. The distribution of birthdays should be roughly uniform, meaning each birthday had an equal chance of being chosen.
7. When it doesnt happen lol. But also I think fairness means total randomness, which in some cases may have clustering.

# Student responses

---

1. I don't think it is ever completely fair, sometimes it's more fair than other times cause if the cause is important
2. If everyone had an equal likelihood of being chosen
3. Im not sure that there really is a fair way to do a draft for war
4. No because each person doesn't have the same probability to be drawn?
5. No, and I don't know because it's randomly selection
6. not sure

# Student responses

---

1. Fairness means that each data should have the same chance of getting drawn out to the draft
2. I think the draft is fair if the people chosen are not chosen due to their race and background but instead a fully random selection. Obviously exclude citizens who have any medical and health issues
3. If there are more people born on certain days of the year, there will be more or less drafted corresponding to different days.
4. it means when we are manipulating data, we make sure how to filter it was not benefit towards certain groups

# Student responses

---

1. equal
2. It mean that no external otucome willa effect and eveyoen will have euqal chance desrite hetier birht date
3. Legal age, healthy, out of everybody no discrimination
4. make sure every date in it instead of missing a date due to some reasons

# Student responses

---

1. birth dates chosen completely at random
2. For the draft to be fair, sampling should be completely random.
3. Random



- Make a scatterplot of ``Rank`` against ``Day`` using all birthdays in ``draft``. - Group by ``Month`` and calculate ``mean_rank``, the mean of ``Rank``, and ``n_dates``, the number of birthdays. - Save the monthly summary as ``draft_months`` and display it. - Label the scatterplot axes so readers know that smaller ranks mean earlier draft priority.

---

72 anonymous responses

Other responses

72

# Student responses

---

1. # 1 & 4. Make a scatterplot with labeled axes indicating smaller ranks mean earlier priority draft |> ggplot(aes(x = Day, y = Rank)) + geom\_point() + labs( x = "Day of the Year", y = "Draft Rank", title = "Draft Priority by Day", caption = "Smaller ranks mean earlier draft priority." ) # 2 & 3. Group by Month, calculate summary statistics, save as draft\_months, and display draft\_months <- draft |> group\_by(Month) |> summarise( mean\_rank = mean(Rank), n\_dates = n() ) draft\_months
2. # Scatterplot draft |> ggplot(aes(x = Day, y = Rank)) + geom\_point() + labs( x = "Day of Year", y = "Draft Rank (smaller rank = earlier draft priority)", title = "Vietnam War Draft Rank by Birthday" ) draft\_months <- draft |> group\_by(Month) |> summarise( mean\_rank = mean(Rank), n\_dates = n() ) draft\_months

# Student responses

---

1. `draft |> ggplot(aes(x = Day, y = Rank)) + geom_point() + labs( title = "Vietnam Draft Lottery Rank by Birthday", x = "Day of the Year", y = "Draft Rank (Lower = Earlier Priority)" ) draft_months <- draft |> group_by(Month) |> summarize( mean_rank = mean(Rank), n_dates = n() ) draft_months`
2. `draft |> ggplot(aes(x = Day, y = Rank)) + geom_point() + labs( x = "Birthday (day of year, 1 = Jan 1)", y = "Draft rank (1 = called first)" ) draft_months <- draft |> group_by(Month) |> summarize(mean_rank = mean(Rank), n_dates = n()) draft_months`
3. `draft |> ggplot(aes(x = Day, y = Rank)) + geom_point() + labs( x = "Day of year", y = "Draft rank (smaller = earlier priority)" ) draft_months <- draft |> group_by(Month) |> summarize( mean_rank = mean(Rank, na.rm = TRUE), n_dates = n() ) draft_months`

# Student responses

---

1. `draft |> ggplot(aes(x = Day, y = Rank)) + geom_point() + labs( x = "Day of Year", y = "Draft Rank (smaller = earlier priority)" ) draft_months <- draft |> group_by(Month) |> summarize( mean_rank = mean(Rank), n_dates = n() ) draft_months`
2. `draft |> ggplot(aes(x = Day, y = Rank)) + geom_point() + geom_smooth(method="lm") + labs( title = "Draft Rank by Birthday", x = "Day of the Year (smaller rank = earlier draft priority)", y = "Draft Rank" ) draft_months <- draft |> group_by(Month) |> summarize( meanrank = mean(Rank), ndates = n() ) draft_months`
3. `draft |> ggplot(aes(x = Day, y = Rank)) + geom_point(alpha = 0.4) + labs( title = "Draft lottery rank by birthday", x = "Day of year (birthday)", y = "Draft rank (smaller = drafted earlier)" ) # Monthly summary draft_months <- draft |> group_by(Month) |> summarize( mean_rank = mean(Rank), n_dates = n(), .groups = "drop" ) draft_months`
4. `draft |> ggplot + geom_point(aes(x = Day, Rank))`
5. `draft |> ggplot(aes(x = Month, Rank)) + geom_boxplot()`

# Student responses

---

1. `draft |> ggplot(draft, aes(x = Day, y = Rank)) + geom_point() + labs( x = "Day of month", y = "Draft rank (smaller ranks = earlier draft priority)" ) draft_months <- draft |> group_by(Month) |> summarize( mean_rank = mean(Rank, na.rm = TRUE), n_dates = n() ) draft_months`
2. `draft |> ggplot(aes(x = Day, Rank)) + geom_boxplot()`
3. `draft |> ggplot(aes(x = Day, y = Rank)) + geom_point() + geom_smooth(method = "lm", se = FALSE) + labs( x = "Day of month", y = "Draft priority rank (smaller = earlier priority)" )` Then make the monthly summary: `draft_months <- draft |> group_by(Month) |> summarize( mean_rank = mean(Rank, na.rm = TRUE), n_dates = n() ) draft_months`
4. `draft |> ggplot(aes(x = Day, y = Rank)) + geom_point() + labs( x = "Day of month", y = "Draft priority rank (smaller = earlier)" ) draft_months <- draft |> group_by(Month) |> summarize( mean_rank = mean(Rank, na.rm = TRUE), n_dates = n() )`

# Student responses

---

1. `draft |> ggplot(aes(x = Day, y = Rank)) + geom_point() + labs( x = "Day of month", y = "Draft rank (smaller = earlier priority)" ) # Then make the monthly summary: draft_months <- draft |> group_by(Month) |> summarize( mean_rank = mean(Rank, na.rm = TRUE), n_dates = n() ) draft_months`
2. `draft |> ggplot(aes(x = Month, Rank)) + geom_boxplot()`
3. `draft |> ggplot() + geom_point(aes(x = Day, y = Rank)) draft_months <- draft |> group_by(Month) |> summarize( mean_rank = mean(Rank), n_dates = n() )`
4. `draft |> ggplot(aes(x = Day, Rank)) + geom_point() + geom_smooth(method = "lm")`
5. `draft |> ggplot(aes(x = Month, y = Rank)) + geom_boxplot() + geom_smooth(method = "lm")`
6. `draft |> ggplot((aes(x=Month, Rank))) + geom_boxplot()`
7. `draft |> ggplot() + geom_point(aes(x = Day, Rank)) + geom_smooth(method = "lm")`
8. `draft |> ggplot() + geom_point(aes(x= Day, Rank))`
9. `draft |> ggplot(aes(x = Day, Rank)) + geom_point() + geom_smooth(method = "lm")`

# Student responses

---

1. `draft |> ggplot(aes(x = Day, Rank)) + geom_point() + geom_smooth(method = "lm")`
2. `draft |> ggplot(aes(x = Day, Rank)) + geom_point() + geom_smooth(method = "lm")`
3. `draft |> ggplot(aes(x = Day, Rank)) + geom_point() + geom_smooth(method = "lm")`
4. `draft |> ggplot(aes(x = Day,y= Rank)) + geom_point() + geom_smooth(method = "lm") + labs( x = "Day of month", y = "Draft rank (smaller = earlier priority)" )`  
`draft_months <- draft |> group_by(Month) |> summarize( mean_rank = mean(Rank), n_dates = n() )`  
`draft_months`  
`draft |> ggplot(aes(x = Month, Rank)) + geom_boxplot()`
5. `draft |> ggplot(aes(x = Month, Rank)) + geom_boxplot()`
6. `draft |> ggplot(aes(x = Month, Rank)) + geom_boxplot()`
7. `draft |> ggplot(aes(x = Month, Rank)) + geom_boxplot()`
8. `draft |> ggplot(aes(x = Month, Rank)) + geom_boxplot()`
9. `draft |> ggplot(aes(x = Month, Rank)) + geom_boxplot()`

# Student responses

---

1. `draft |> ggplot(aes(x = Month, Rank)) + geom_boxplot()`
2. `draft |> ggplot(aes(x = Month, Rank)) + geom_boxplot()`
3. `draft |> ggplot(aes(x = Month, Rank)) + geom_boxplot()`
4. `draft |> ggplot(aes(x = Month, Rank)) + geom_boxplot()`
5. `draft |> ggplot(aes(x = Month, Rank)) + geom_boxplot()`
6. `draft |> ggplot(aes(x = Month, Rank)) + geom_boxplot()`
7. `draft |> ggplot(aes(x = Month, Rank)) + geom_boxplot()`
8. `draft |> ggplot(aes(x = Month, Rank)) + geom_boxplot()`
9. `draft |> ggplot(aes(x = Month, Rank)) + geom_boxplot()`



# Student responses

---

1. `draft |> ggplot(aes(x = Month, Rank)) + geom_boxplot()`
2. `draft |> ggplot(aes(x = Month, Rank)) + geom_boxplot()`
3. `draft |> ggplot(aes(x = Month, Rank)) + geom_boxplot()`
4. `draft |> ggplot(aes(x = Month, Rank)) + geom_boxplot()`
5. `draft |> ggplot(aes(x = Month, y = Rank)) + geom_boxplot()`
6. `draft |> ggplot(aes(x = Month, y = Rank)) + geom_boxplot()`
7. `draft |> ggplot(aes(x = Month, y = Rank)) + geom_boxplot()`
8. `draft |> ggplot(aes(x = Month, y = Rank)) + geom_boxplot()`
9. `draft |> ggplot(aes(x = Month, y = Rank)) + geom_boxplot()`

# Student responses

---

1. `draft |> ggplot(aes(x = Month, y = Rank)) + geom_boxplot()`
2. `draft |> ggplot(aes(x = Month, y = Rank)) + geom_boxplot()`
3. `draft |> ggplot(aes(x = Month, y = Rank)) + geom_boxplot() + geom_smooth(method = "lm")`
4. `draft |> ggplot(aes(x = Month, y = Rank))+ geom_boxplot()`
5. `draft |> ggplot(aes(x= Day,y=Rank))+ geom_point()+ geom_smooth(method = "lm")`
6. `draft |> ggplot(aes(x= Month, Rank)) + geom_boxplot()`
7. `draft |> ggplot(aes(x=Day,Rank)) + geom_boxplot()`
8. `draft |> ggplot(aes(x=Day,Rank))+ geom_point()+ geom_smooth(method = "lm")`
9. `draft |> ggplot(aes(x=Month, Rank)) + geom_boxplot()`

# Student responses

---

1. `draft |> ggplot(aes(x=Month, Rank)) + geom_boxplot()`
2. `draft |> ggplot(aes(x=Month, Rank)) + geom_boxplot()`
3. `draft |> ggplot(aes(x=Month,Rank)) + geom_boxplot()`
4. `draft |> ggplot(aes(x=Month,y=Rank)) + geom_boxplot()`
5. `draft |> head()`
6. `draft |> head() library(dplyr) library(ggplot2) ggplot(draft, aes(x = Day, y = Rank)) + geom_point() + labs( x = "Day of year", y = "Draft rank (smaller = earlier priority)" ) draft_months <- draft %>% group_by(Month) %>% summarise( mean_rank = mean(Rank, na.rm = TRUE), n_dates = n() ) draft_months`
7. `draft |> head() ggplot(draft, aes(x = Day, y = Rank)) + geom_point() + labs( x = "Day of year", y = "Draft rank (smaller = earlier priority)" ) draft_months <- draft |> group_by(Month) |> summarize( mean_rank = mean(Rank), n_dates = n() ) draft_months`
8. `draft|> ggplot(aes(x = Month, y = Rank))+geom_boxplot()`
9. `draft|>ggplot(aes(x=Month, y=Rank))+ geom_boxplot()`

# Student responses

---

1. `ggplot(draft, aes(x = Day, y = Rank)) + geom_point() + labs( x = "Day of year (birthday)", y = "Draft rank (smaller = earlier priority)" ) draft_months <- draft |> group_by(Month) |> summarize( mean_rank = mean(Rank, na.rm = TRUE), n_dates = n() ) draft_months`
2. `ggplot(draft, aes(x = Day, y = Rank)) + geom_point() + labs( x = "Day of year", y = "Draft rank (smaller = earlier priority)" ) draft_months <- draft %>% group_by(Month) %>% summarise( mean_rank = mean(Rank, na.rm = TRUE), n_dates = n() ) draft_months`
3. `library(dplyr) library(ggplot2) ggplot(draft, aes(x = Day, y = Rank)) + geom_point() + labs( title = "Draft priority by birthday", x = "Birthday (day of the year)", y = "Draft rank (smaller = earlier priority)" ) + theme_minimal() draft_months <- draft |> group_by(Month) |> summarise( mean_rank = mean(Rank, na.rm = TRUE), n_dates = n(), .groups = "drop" ) draft_months`

# Student responses

---

1. 

```
library(dplyr) library(ggplot2) ggplot(draft, aes(x = Day, y = Rank)) + geom_point() + labs( x = "Day of year", y = "Draft rank (smaller = earlier priority)" ) +  
  geom_smooth(method = "lm") draft_months <- draft %>% group_by(Month) %>% summarise( mean_rank = mean(Rank, na.rm = TRUE), n_dates = n() )  
draft_months
```

**- Use all birthdays in `draft`, mapping `Month` to x and `Rank` to y. - Add `geom\_boxplot()` to make one boxplot per month, ordered January through December. - Label the axes so readers know that smaller lottery numbers mean earlier call-up.**

---

0 anonymous responses

No responses were submitted.

**- Calculate the mean `Rank` for each half of the year using `group\_by()` and `summarize()`. - Run a two-sample t-test comparing `Rank` between the two halves of the year. - In a code comment, report both means and explain whether the difference is significant at the 5% level and which half faced earlier call-up.**

---

67 anonymous responses

Other responses

67

# Student responses

---

1. `draft_halves %>% group_by(first_half) %>% summarize(mean_rank = mean(Rank, na.rm = TRUE)) t.test(Rank ~ first_half, data = draft_halves) # Mean Rank for Jan-Jun (first_half = TRUE): __ # Mean Rank for Jul-Dec (first_half = FALSE): __ # If the t-test p-value is less than 0.05, the difference is statistically significant at the 5% level. # The half with the lower mean Rank faced earlier call-up.`
2. `draft_halves %>% group_by(first_half) %>% summarize(mean_rank = mean(Rank, na.rm = TRUE)) t.test(Rank ~ first_half, data = draft_halves) # Mean Rank for Jan-Jun (first_half = TRUE): 206 # Mean Rank for Jul-Dec (first_half = FALSE): 161 # The t-test p-value is less than 0.05, so the difference is statistically significant at the 5% level. # The half with the lower mean Rank faced earlier call-up.`
3. `draft_halves <- draft |> mutate( first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) t.test(Rank~first_half, data = draft_halves)`



# Student responses

---

1. `draft_halves <- draft |> mutate( first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sept", "Oct", "Nov", "Dec")) t.test(Rank ~ first_half, data = draft_halves)`
2. `draft_halves <- draft |> mutate( first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) ) draft_halves |> group_by(first_half) |> summarize(mean_rank = mean(Rank)) t.test(Rank ~ first_half, data = draft_halves) # The mean Rank was 206 for Jan-Jun and 161 for Jul-Dec. # The difference was statistically significant at the 5% level because the p-value was less than 0.05. # Jul-Dec had an earlier call-up on average.`
3. `draft_halves <- draft |> mutate( first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) ) t.test(Rank ~ first_half, data = draft_halves)`
4. `draft_halves <- draft |> mutate( first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) ) t.test(Rank ~ first_half, data = draft_halves) draft_halves |> head()`

# Student responses

---

1. `draft_halves <- draft |> mutate( first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun") ) t.test(Rank ~ first_half, data = draft_halves)`
2. `draft_halves <- draft |> mutate( first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun") ) t.test(Rank ~ first_half, data = draft_halves)`
3. `draft_halves <- draft |> mutate( first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun"), ) draft_halves |> head() draft_halves |> group_by(first_half) |> summarize( mean_rank = mean(Rank) )`
4. `draft_halves <- draft |> mutate( first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) draft_halves`
5. `draft_halves <- draft |> mutate( first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) draft_halves |> head() t.test(Rank ~ first_half, data = draft_halves)`
6. `draft_halves <- draft |> mutate( first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) draft_halves |> slice_sample(n=10) t.test(Rank ~ first_half, data = draft_halves)`

# Student responses

---

1. `draft_halves <- draft |> mutate( first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) t.test(Rank~first_half,data=draft_halves)`
2. `draft_halves <- draft |> mutate( first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) t.test(Rank ~ first_half, data = draft_halves) # the mean in group false is 160 and #206 is true. The difference is significant at the 5% level.`
3. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) t.test(Rank ~ first_half, data = draft_halves)`
4. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) t.test(Rank ~ first_half, data = draft_halves)`
5. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) draft_halves |> head() t.test(Rank ~ first_half, data = draft_halves)`
6. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) t.test(Rank ~ first_half, data = draft_halves)`

# Student responses

---

1. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) t.test(Rank ~ first_half, data = draft_halves)`
2. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) t.test(Rank ~ first_half, data = draft_halves)`
3. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) t.test(Rank ~ first_half, data = draft_halves)`
4. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) t.test(Rank ~ first_half, data = draft_halves)`
5. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) t.test(Rank ~ first_half, data = draft_halves) #There is a difference in mean rank among the first half and second half. There is a bias meaning that draftees who have a birthday in the second half of the year are more likely to be chosen than the draftees with a birthday in the first half of the year.`

# Student responses

---

1. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun"))` # Mean rank for each half `draft_halves |> group_by(first_half) |> summarize(mean_rank = mean(Rank), n = n(), .groups = "drop")` # Two-sample t-test `t.test(Rank ~ first_half, data = draft_halves)` # `first_half = TRUE` (Jan-Jun) mean rank about 206.8, `first_half = FALSE` (Jul-Dec) about 160.9. # The t-test p-value is well below 0.05, so the difference is significant at the 5% level. # The second half of the year (Jul-Dec) has the lower mean rank, so those birthdays faced # earlier call-up.

# Student responses

---

1. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun"))` `draft_halves |> group_by(first_half) |> summarize(mean_rank = mean(Rank) )` `t.test(Rank ~ first_half, data = draft_halves)` # The average lottery number was about 206.3 for the first six months # and 160.9 for the last six months. The p-value is much smaller than 0.05, # so the difference is statistically significant at the 5% level. # Since a smaller lottery number means earlier draft priority, people born # in the last six months faced earlier call-up.
2. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun"))` `draft_halves |> group_by(first_half) |> summarize(mean_rank = mean(Rank, na.rm = TRUE))` `t.test(Rank ~ first_half, data = draft_halves)`

# Student responses

---

1. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) draft_halves |> group_by(first_half) |> summarize(mean_Rank = mean(Rank, na.rm = TRUE)) t.test(Rank ~ first_half, data = draft_halves) # Very significant Evidence since there there is a lot of mean and also 5 percent`
2. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) draft_halves |> group_by(first_half) |> summarize(mean_rank = mean(Rank)) t.test(Rank ~ first_half, data = draft_halves) # Mean rank for the first half of the year = 161 # Mean rank for the second half of the year = 206 # The p-value is 3.422e-05, so the difference is statistically significant # at the 5% level. The half with the smaller mean rank faced earlier call-up.`
3. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) draft_halves |> head()`
4. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) draft_halves |> head()`

# Student responses

---

1. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) draft_halves |> head()`
2. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) draft_halves |> head()`
3. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) draft_halves |> head() draft_halves %>% group_by(first_half) %>% summarize(mean_rank = mean(Rank, na.rm = TRUE)) t.test(Rank ~ first_half, data = draft_halves) # Mean Rank for Jan-Jun: __ # Mean Rank for Jul-Dec: __ # If the t-test p-value is less than 0.05, the difference is statistically significant at the 5% level. # The half with the lower mean Rank faced earlier call-up.`



# Student responses

---

1. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) draft_halves |> head()` `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) draft_halves |> group_by(first_half) |> summarise(mean_rank = mean(Rank))`  
`t.test(Rank ~ first_half, data = draft_halves)` #The first half of the year has a lower mean rank compared to the second half. # At the 5% significance level ( $p < 0.05$ ), the difference is/is not statistically significant. # Since smaller ranks mean earlier draft priority, the first half faced earlier call-up.

# Student responses

---

1. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) draft_halves |> head()` `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) draft_halves |> group_by(first_half) |> summarize(mean_rank = mean(Rank)) t.test(Rank ~ first_half, data = draft_halves) # Compare the two mean ranks above. # If p-value < 0.05, the difference is statistically significant. # The half with the smaller mean Rank faced earlier call-up.`
2. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) t.test(Rank ~ first_half, data = draft_halves)`
3. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) t.test(Rank ~ first_half, data = draft_halves)`
4. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) t.test(Rank ~ first_half, data = draft_halves)`

# Student responses

---

1. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) t.test(Rank ~ first_half, data = draft_halves)`
2. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) t.test(Rank ~ first_half, data = draft_halves)`
3. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) t.test(Rank ~ first_half, data = draft_halves)`
4. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) t.test(Rank ~ first_half, data = draft_halves)`
5. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) t.test(Rank ~ first_half, data = draft_halves)`
6. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) t.test(Rank ~ first_half, data = draft_halves)`

# Student responses

---

1. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) t.test(Rank ~ first_half, data = draft_halves) #The means are 160.9239 and 206.3242. The p-value is close to 0 which is less than 0.05, so we reject the null hypothesis`
2. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) t.test(Rank ~ first_half, data = draft_halves) #we screwed if born in second half of year`
3. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) t.test(Rank ~ first_half, data = draft_halves) # means are 160.93 for F and 206.32 for TRUE. The difference is real, with a p-value being so minimal`
4. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) t.test(Rank ~ first_half, data = draft_halves) draft_halves |> head()`
5. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) t.test(Rank ~ first_half, data=draft_halves)`

# Student responses

---

1. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) t.test(Rank ~ first_half, data=draft_halves) #The means are 160 and 206, and p-value is not significant. The second half faced an earlier call-up.`
2. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) t.test(Rank~first_half, data = draft_halves)`
3. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) draft_halves %>% group_by(first_half) %>% summarize(mean_rank = mean(Rank, na.rm = TRUE)) t.test(Rank ~ first_half, data = draft_halves) #206 <- second half mean, first half mean: 161, pval is < 0.05 so this is significant, reject the null`
4. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) draft_halves |> group_by(first_half) |> summarize(mean_rank = mean(Rank)) t.test(Rank ~ first_half, data = draft_halves) # Means: 160.9239 and 206.3242 respectively, we do see a statistically significant difference # draft_halves |> head()`

# Student responses

---

1. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) t.test(Rank ~ first_half, data = draft_halves)`
2. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) t.test(Rank ~ first_half, data = draft_halves)`
3. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) t.test(Rank ~ first_half, data = draft_halves)`
4. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) t.test(Rank ~ first_half, data = draft_halves)` # Mean Rank was 160.9239 for Jan–Jun and 206.3242 for Jul–Dec. # Because the p-value was 3.422e-05 , the difference was / was not significant at the 5% level. # The \_\_\_\_ half had earlier call-up because it had the lower average Rank.
5. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) t.test(Rank ~ first_half, data=draft_halves)`

# Student responses

---

1. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) group_by(first_half) |> summarize(mean_rank = mean(Rank))  
t.test(Rank ~ first_half, data = draft_halves) draft_halves |> head()`
2. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) |> group_by(first_half) |> summarize(mean_rank =  
mean(Rank)) t.test(Rank ~ first_half, data = draft_halves) t.test(draft_halves, rank) draft_halves |> head()`
3. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) |> group_by(first_half) |> summarize(mean_rank =  
mean(Rank)) t.test(Rank ~ first_half, data = draft_halves) # in group false, the mean is 160.9239, and the mean in group TRUE is 206.32. The difference is  
significant at 5% level`
4. `draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) |> group_by(Rank) |> summarize( mean_rank = mean(Rank) )  
draft_halves |> head() t.test(Rank ~ first_half, data = draft_halves)`

# Student responses

---

1. `draft_halves |> group_by(first_half) |> summarize(mean_rank = mean(Rank)) t.test(Rank ~ first_half, data = draft_halves)` # The average lottery number was 206 for January–June and 161 for July–December. # The difference was statistically significant ( $p < 0.05$ ), meaning people # born in the last six months faced earlier draft call-ups.
2. `draft_halves |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) |> t.test(Rank ~ first_half, data = _)`
3. `draft_halves <- draft |> mutate( first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun") ) t.test(Rank ~ first_half, data = draft_halves)`



# Student responses

---

1. 

```
library(dplyr) draft_halves <- draft |> mutate(first_half = Month %in% c("Jan", "Feb", "Mar", "Apr", "May", "Jun")) draft_halves |> group_by(first_half) |> summarise(mean_rank = mean(Rank), .groups = "drop") result <- t.test(Rank ~ first_half, data = draft_halves) result # Generate a comment reporting the results: first_mean <- with(draft_halves, mean(Rank[first_half])) last_mean <- with(draft_halves, mean(Rank[!first_half])) cat(sprintf( "# January–June mean: %.2f; July–December mean: %.2f\n# p = %.4g: %s at the 5%% level.\n# %s had the lower average rank and therefore earlier call-up priority.\n", first_mean, last_mean, result$p.value, ifelse(result$p.value < 0.05, "statistically significant", "not statistically significant"), ifelse(first_mean < last_mean, "January–June", "July–December") ))
```

**Explain who was disadvantaged by the lottery and how the mixing procedure produced the pattern. What pattern in the plots supports your conclusion?**

---

0 anonymous responses

No responses were submitted.