

Creating data pipelines

Week 2, Lecture 4

<https://live.ds306.org>

September 10, 2026

Where we left off

```
library(tidyverse)
theme_set(theme_minimal(base_size = 13, base_family = "sans"))

load("a2weather.RData")
a2weather <- as_tibble(a2weather)
survey <- read_csv("class_survey.csv", show_col_types = FALSE)
```

- Last time we summarized daily temperatures and counted extreme precipitation days.
- Today we'll use the same table verbs to answer two new questions.
- The main task today is learning how to break down data problems into sequences of operations with table verbs: `filter()`, `mutate()`, `summarize()`, `arrange()`, `group_by()`, and `select()`.

What we'll practice

- Plan the intermediate tables before writing a long pipeline.
- Filter observations, summarize groups, then calculate with the summaries.
- Construct a contingency table and test for association with `chisq.test()`.

Today's questions

1. Which day would you choose for a club picnic on the Diag?
2. What food should we order for the picnic? Do preferences differ between groups?

When should we hold a picnic on the Diag?



Here's the event

- Your student organization wants to hold an afternoon picnic on the Diag.
- There is no tent or covered space.
- Assume attendees and space are available any day from May through October. (Pretend you are here for the summer!)
- Our job is to use data to pick the best day to hold the picnic.

Make a plan before opening R

Before looking at the data, pick a date that you think would be best for the picnic, and explain your initial choice. What weather would make it enjoyable or force you to cancel? Describe how you would use the weather records to evaluate your choice.

Turning your idea into an analysis

1. Decide what matters to the people attending the picnic.
2. Identify measurements that could help us evaluate it.
3. Decide how to compare dates using those measurements.
4. Perform the analysis.
5. Report your findings.

What can these records tell us?

```
glimpse(a2weather)
```

- tmin and tmax are the daily low and high, in Celsius.
- prcp is daily precipitation, in millimeters.
- These are daily measurements, not the weather during the picnic.

Two simple rules

1. Pick the calendar date that was rainy in the fewest of the past 30 years.
2. Pick the calendar date whose average temperature over those years is closest to 72°F.

We'll work together on building each rule into a pipeline. Then you'll propose your own rule.

Compare the same date across years

- `year(date)`, `month(date)`, and `day(date)` give the year, month, and day of the month.
- We'll compare dates from May 1 through October 31.

Keep the relevant days

As we saw, the weather is changing. We'll use the period 1996–2025 as the basis for our analysis.

```
event_days <- a2weather |>
  mutate(year = year(date), month = month(date), day = day(date)) |>
  filter(year >= 1996, year <= 2025, month >= 5, month <= 10) |>
  select(date, year, month, day, tmin, tmax, prcp)
```

Rule 1: rain in the fewest years

What do we need to do?

1. Decide what counts as a rainy day.
2. For each calendar date, count how many of the 30 years were rainy.
3. Sort those counts from smallest to largest, or plot them.

Mark the rainy days

For this example, call a day **rainy** if it has at least 1 mm of precipitation.

```
rain_days <- event_days |> mutate(rainy = prcp >= 1)  
rain_days |> select(date, month, day, prcp, rainy) |> head()
```

- The new column contains TRUE, FALSE, or NA.
- Is 1mm a reasonable choice?

Count rainy years for each date

```
rain_by_date <- rain_days |>
  group_by(day) |>
  summarize(
    n_years = sum(!is.na(prcp)),
    n_rainy = sum(rainy, na.rm = TRUE),
    .groups = "drop"
  )
```

- `sum(rainy)` counts TRUE values: each contributes 1.

What went wrong?

We only used 30 years of data, but `n_years` is 180 in most rows.
Explain what went wrong and how you would fix the code.

Keep the month and day together

The first attempt combined May 10, June 10, and so on. We need a separate group for each calendar date.

```
rain_by_date <- rain_days |>
  group_by(month, day) |>
  summarize(
    n_years = sum(!is.na(prcp)),
    n_rainy = sum(rainy, na.rm = TRUE),
    .groups = "drop"
  )

head(rain_by_date)
```

`.groups = "drop"` leaves the result ungrouped after the summary.

Are the counts comparable?

```
rain_by_date |> filter(n_years != 30)
```

No rows: every calendar date has a recorded precipitation value in all 30 years.

Sort from fewest rainy years to most

```
rain_ranking <- rain_by_date |>  
  arrange(n_rainy)  
  
head(rain_ranking)
```

- August 31 and September 17 tie for the fewest rainy days.

Put the rain rule into one pipeline

We can combine the steps without saving each intermediate table.

```
rain_ranking <- event_days |>
  mutate(rainy = prcp >= 1) |>
  group_by(month, day) |>
  summarize(
    n_years = sum(!is.na(prcp)),
    n_rainy = sum(rainy, na.rm = TRUE),
    .groups = "drop"
  ) |>
  arrange(n_rainy)
```

Visualizing the result

```
rain_ranking |>
  ggplot(aes(x = make_date(2025, month, day), y = n_rainy)) +
  geom_point() + labs(x = "Date", y = "Number of Rainy Years")
```

Rule 2: an average closest to 72°F

What do we need to do?

1. Estimate each day's average temperature and convert to Fahrenheit.
2. Average those temperatures for each calendar date across the 30 years.
3. Calculate how far each average is from 72°F.
4. Sort from smallest distance to largest.

Estimate each day's average temperature

There is no tavg column. We'll use the midpoint of the daily low and high.

```
temperature_days <- event_days |>
  mutate(
    temp_c = (tmin + tmax) / 2,
    temp_f = temp_c * 9 / 5 + 32
  )

temperature_days |> select(date, tmin, tmax, temp_f) |> head()
```

Average the temperatures for each date

```
temperature_by_date <- temperature_days |>
  group_by(month, day) |>
  summarize(
    n_years = sum(!is.na(temp_f)),
    mean_temp = mean(temp_f),
    .groups = "drop"
  )
```

Why are some averages missing?

```
temperature_by_date |>  
  filter(is.na(mean_temp)) |>  
  head(3)
```

Each of these dates has 29 recorded temperatures. Why is its mean still NA? Explain what went wrong and how to fix it.

Calculate the mean of the recorded temperatures

```
temperature_by_date <- temperature_days |>
  group_by(month, day) |>
  summarize(
    n_years = sum(!is.na(temp_f)),
    mean_temp = mean(temp_f, na.rm = TRUE),
    .groups = "drop"
  )

temperature_by_date |> filter(month == 8, day == 9)
```

`mean( )` returns NA if any input is missing. With `na.rm = TRUE`, we get the mean of the 29 recorded temperatures for August 9, not all 30 years. We have not recovered the missing value.

Calculate the distance from 72°F

```
temperature_by_date <- temperature_by_date |>
  mutate(distance = mean_temp - 72)

temperature_by_date |> select(month, day, mean_temp, distance) |> head()
```

We use `mutate()` after `summarize()` because we need the mean for each calendar date first.

Sort from closest to farthest

```
temperature_ranking <- temperature_by_date |>  
  arrange(distance)  
  
head(temperature_ranking)
```

Look at the dates at the top of this ranking. Did we find averages closest to 72°F? Explain what went wrong and how you would fix it.

Calculate distance in either direction

- Sorting `mean_temp` - 72 put the coldest dates first, not the closest to 72°F.
- `abs( )` gives the absolute value: 68°F is 4 degrees away; 75°F is 3 degrees away.

```
temperature_by_date <- temperature_by_date |>
  mutate(distance = abs(mean_temp - 72))

temperature_by_date |> select(month, day, mean_temp, distance) |> head()
```

Try the ranking again

```
temperature_ranking <- temperature_by_date |>  
  arrange(distance)  
  
head(temperature_ranking)
```

- This rule ignores rain and how much temperatures vary around the average.

Put the temperature rule into one pipeline

```
temperature_ranking <- event_days |>
  mutate(temp_f = ((tmin + tmax) / 2) * 9 / 5 + 32) |>
  group_by(month, day) |>
  summarize(
    n_years = sum(!is.na(temp_f)),
    mean_temp = mean(temp_f, na.rm = TRUE),
    .groups = "drop"
  ) |>
  mutate(distance = abs(mean_temp - 72)) |>
  arrange(distance)
```

Averaging first, then taking the distance, implements the rule we chose.

Now propose your own rule

The rain rule can choose a cold date; the temperature rule can choose a wet one. Both rules only look at averages and ignore variability. Propose a rule that addresses more than one concern for this picnic. What would you calculate for each observation, and what would you summarize across years for each calendar date? Does your rule make any tradeoffs?

Build your analysis in five steps

1. Score each day according to your criterion.
2. Group the same calendar date across years.
3. Summarize day, while handling missing values.
4. Rank the days.
5. In an R comment, report your recommendation.

Try your rule

Start from `event_days`. Write your recommendation in a comment below your code.

```
event_days |> head()  
  
# Recommendation:
```

- Define a weather-based score for each day.
- Group the same calendar date across years, using month and day.
- Summarize each date while handling missing values.
- Rank the calendar

What would you recommend?

Choose a date using your rule. Give a number from your results that supports your choice. How does it compare with the two simple rules, and what would you still want to know before booking the picnic?

Would another reasonable choice change our answer?

Choose one part of your rule that another group might reasonably disagree with. Predict how changing it would affect the ranking, then rerun the analysis. Does your recommendation change? What would you now tell the club?

Two orders of operations, two questions

```
event_days |>
  filter(tmax >= 27) |>
  group_by(month, day) |>
  summarize(mean_high = mean(tmax), .groups = "drop")
```

```
event_days |>
  group_by(month, day) |>
  summarize(mean_high = mean(tmax, na.rm = TRUE), .groups = "drop") |>
  filter(mean_high >= 27)
```

Both pipelines run. What question does each answer? Explain which days contribute to each mean, rather than just describing the lines of code.

What should we order for the picnic?

Would one menu suit everyone?

- We have a date in mind. Now we need to choose the food.
- We'll use the survey data to inform our picnic menu choices.

Predict, then plan the comparison

Would you expect the same food to be most popular in both groups?
What would you calculate to decide whether one menu would suit both? Explain what an overall ranking of food choices might leave out.

Food responses

```
food_data <- survey |>
  select(favorite_food, birth_continent) |>
  filter(!is.na(favorite_food), !is.na(birth_continent))

food_data |>
  group_by(birth_continent) |>
  summarize(n_respondents = n())
```

What is most popular overall?

```
food_data |>
  group_by(favorite_food) |>
  summarize(n_votes = n()) |>
  arrange(desc(n_votes))
```

Sushi has 36 votes, followed by burritos with 22.

Count food choices within each group

```
food_counts <- food_data |>
  group_by(birth_continent, favorite_food) |>
  summarize(n = n(), .groups = "drop")

food_counts |> filter(favorite_food == "Sushi")
```

There are more sushi votes from respondents reporting North America, but that group is much larger.

Using count ()

```
food_data |>  
  count(birth_continent, favorite_food)
```

Counting is so common that we have a verb count () for it.

Arrange counts in a contingency table

A **contingency table** displays counts for combinations of categorical variables.

```
food_table <- food_data |>
  select(favorite_food, birth_continent) |>
  table()
```

Rows are foods; columns are birth-continent groups.

Joint distribution: food and birth continent

What fraction of respondents chose sushi and reported Asia?

```
joint <- prop.table(food_table)
```

- Every count is divided by 98.
- The Sushi / Asia entry is 10/98, or about 10.2%.
- All 16 entries together sum to 1.

Marginal distribution: food alone

What fraction chose sushi, regardless of birth continent?

```
food_marginal <- margin.table(joint, margin = 1)
```

- Add across the two continent columns for each food.
- `margin = 1` keeps the food rows. This is **marginalizing** over birth continent.
- For sushi, $(10 + 26)/98 = 36/98$, or about 36.7%.

Conditional distribution: food within a group

Among respondents who reported Asia, what percentage chose sushi?

```
100 * prop.table(food_table, margin = 2)
```

- Our groups are in the columns, so `margin = 2` divides by each column's total.
- Each column sums to 100%.
- The percentages are $10/18 = 55.6\%$ and $26/80 = 32.5\%$.

Without a margin, divide by the whole table's total

```
100 * prop.table(food_table)
```

- Every count is divided by 98, so these are joint percentages.
- For Sushi / Asia, we now have $10/98 = 10.2\%$, not $10/18 = 55.6\%$.
- All 16 entries together sum to 100%, not each column separately.

What would independence mean for the menu?

- Under independence, the probability of choosing each food would be the same in both birth-continent groups.
- Each conditional food distribution would equal the marginal food distribution.
- Let's compare the observed counts with counts expected under independence.

Get both marginal distributions

```
food_marginal <- margin.table(joint, margin = 1)
continent_marginal <- margin.table(joint, margin = 2)
```

```
food_marginal
continent_marginal
```

- The food proportions sum to 1, and so do the continent proportions.
- These are the probabilities we estimate from the survey: for example, 36/98 for sushi and 18/98 for Asia.

Why multiply the marginals?

- The fraction of respondents who reported Asia is $18/98$.
- Under independence, the probability of choosing sushi within that group equals the overall sushi probability, which we estimate as $36/98$.
- We therefore estimate the fraction who reported Asia **and** chose sushi as $(18/98) * (36/98)$.

Under independence, the joint probability is the product of the two marginal probabilities.

Build the product-of-marginals table

`outer ( )` multiplies every entry in the first vector by every entry in the second.

```
expected_joint <- outer(food_marginal, continent_marginal)
round(expected_joint, 3)
```

For Sushi / Asia: $(36/98) * (18/98)$, or about 0.067.

Convert the proportions to expected counts

Multiply every entry by the number of respondents.

```
expected_counts <- sum(food_table) * expected_joint  
round(expected_counts, 1)
```

- For Sushi / Asia, the expected count is about 6.6.
- Fractional expected counts are fine; these are averages under the independence model.
- We round the display, not the values stored in `expected_counts`.

Compare observed and expected counts

Observed counts

```
food_table
```

Expected under independence

```
round(expected_counts, 1)
```

Does a difference rule out independence?

The observed counts differ from the counts we constructed using the product of the marginals. Does this establish that food choice and birth continent are not independent in the population? Explain your reasoning. What else would you need to know before drawing that conclusion?

How much disagreement should surprise us?

- A random sample can produce unequal food percentages even when food choice and birth continent are independent.
- We do not expect the observed and expected counts to match exactly.
- We need to ask how often independent choices would produce disagreement at least as large as ours.
- The chi-square test measures disagreement across the whole table.

Some foods have very few votes

Look at the expected count for Salad / Asia:

```
expected_counts["Salad", "Asia"]
```

- The expected count is about 0.37, even though we have 98 respondents overall.
- The usual chi-square approximation is unreliable when many expected counts are this small.

Test for independence

```
set.seed(306)
chisq.test(
  food_table, simulate.p.value = TRUE, B = 10000
)
```

Interpret the test

Sushi is the most popular choice in both groups, but its share is 55.6% in one and 32.5% in the other. The test of the full table has a p-value around 0.18. What would you recommend ordering? Use the observed choices and explain what the test does, and does not, establish.

What else do we need before ordering?

- A large p-value does not establish that the groups have identical preferences.
- Our class survey does not represent everyone born on either continent.
- A favorite-food vote does not tell us what else someone would happily eat.
- We still need attendance numbers, dietary requirements, and a budget.

Checking an analysis that runs

An answer from an AI assistant

An assistant calculates the percentage choosing sushi within each birth-continent group:

```
food_data |>
  filter(favorite_food == "Sushi") |>
  group_by(birth_continent) |>
  summarize(n_sushi = n()) |>
  mutate(percent = 100 * n_sushi / sum(n_sushi))
```

What do these percentages actually describe? Which votes were removed too soon, and how would you repair the calculation?

Count all votes before filtering

```
food_data |>
  group_by(birth_continent) |>
  summarize(
    n_respondents = n(),
    n_sushi = sum(favorite_food == "Sushi")
  ) |>
  mutate(percent = 100 * n_sushi / n_respondents)
```

This agrees with the sushi percentages within each column of our contingency table.

What to ask while building a pipeline

- Which observations should be included?
- What does one row represent before and after this step?
- Which columns will I need later?
- What belongs in the denominator?
- Does the final table answer the question we started with?

Exit question

Suppose an outdoor event must take place on a day in September through November, and can tolerate cooler weather. Describe how you would change our weather analysis. Which steps would stay the same, and why would adding November require going back to the daily data?

Week 2, Lecture 4

Closed September 10, 2026 at 4:48 PM

14 class questions · 562 anonymous responses · 0 student questions

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

Before looking at the data, pick a date that you think would be best for the picnic, and explain your initial choice. What weather would make it enjoyable or force you to cancel? Describe how you would use the weather records to evaluate your choice.

81 anonymous responses

Other responses	46
Sep 26	6
Late-September date chosen for mild, low-humidity conditions—neither too hot nor cold—and minimal rain. Students intend to compare historical precipitation and temperature to confirm suitability.	
Sep 10	5
Chosen because current/typical early-September weather is mild and agreeable for outdoor plans. Shared concern is canceling for rain; intention to review records for rainfall and temperature patterns.	
Sep 12	5
Favored as a post-start-of-term weekend with warm but not hot conditions and lower rainfall likelihood. Students would use historical precipitation and temperature data to validate the choice.	

Response themes for question 1

Sep 25

4

Late-September Friday chosen for temperate, comfortable conditions around 70–75°F and higher attendance. Main worries are extreme rain or heat; plan to consult past precipitation and temperature data.

Aug 15

3

Selected for late-summer weekend freedom and generally pleasant sunny conditions before fall; expectations of mild temperatures. Reasons emphasize checking historical weather to confirm suitability.

Jun 12

3

Chosen for end-of-week convenience and lighter campus crowds; expected mild early-summer warmth. Concerns focus on rain prompting cancellation; evaluative plans include checking historical precipitation and temperatures.

Response themes for question 1

Jun 20

3

Picked to maximize attendance on a Saturday and for moderate early-summer temperatures. Main rationale is availability; students plan to consult past rainfall and temperature records to assess risk.

Sep 13

3

Selected for a weekend day with comfortable, cooling early-fall weather and minimal scheduling conflicts; students expect pleasant temperatures and low interference from events like football.

Sep 27

3

Chosen as a Sunday late in September for more consistent, slightly cooler fall weather and broader availability; evaluation would rely on past precipitation and daily high/low trends.

Student responses

1. 1st day of the month, the semester is coming to a close and the weather is starting to get nicer so it would be a euphoric activity to have a picnic on this day
2. A nice Saturday when it's not so hot.
3. A Sunday picnic is the best time to have. Also, september 20th is in the peak of fall so the weather should be perfect, not too warm and not too cold.
4. because it is after july 4th so it will not be as crowded
5. Because it is my birthday, and I think it was great weather and no rain
6. because it is my birthday, it is great to go out
7. Because it's near the end of the semester. Many students have already finished the semester and it's a good time for relaxation before they go back to their home. A sunny day would work, so I would check the dataset and filter out those days with no precipitation for reference.
8. During May, it is not too hot but not too cold. I feel like this date would provide the best temperature, excluding other circumstances like rain and wind
9. Early fall likely to have nice weather, only thing to look out for is rain. In the school year can't choose summer

Student responses

1. Early june should be late enough in the year to not have cold weather, but not too hot for summer weather.
2. Early May is warm, but before hotter summer weather hits. It also rains less in late spring, as opposed to now when it is less hot, but fairly rainy.
3. Good weather
4. I chose June 7th. June because it is the beginning of summer and hopefully wont be as hot as July or August. I also chose sunday because hopefully people are les busy that day.
5. I chose this day because of the saying "April Showers bring May Flowers." Rain or other inclimate weather would make the event unenjoyable, so choosing a day mid-May would allow for the time for the April rain to stop. May is typically pretty sunny and not too hot yet.
6. I feel like it's a day that a lot of people will be on campus and the weather would potentially be good. I think it should be sunny and not rainy.
7. I like this day because it is before most midterms. It is also my most free day
8. I personally like the weather to be cooler. Friday is a pretty good day to meet because most people don't have many classes.

Student responses

1. I picked this day because generally, the temperature in mid-September is very nice, between 60 and 79 degrees Fahrenheit. I picked a Friday, as people usually have less work to do than other days. Any weather that is too hot, cold, or a day with too much rain/snow would not be optimal
2. I select this day because the weather predicted in Ann Arbor is predicted to be sunny and warm, but not too hot. People also tend to have classes off this day.
3. I think a Sunday would be best because there are no classes or football games. Weather with rain would force a cancellation.
4. I think that June is a good month because it is nice weather but not too hot yet. Also, I think Friday is a good day because people usually are more available and more open to going to different events.
5. I think this day is pretty good. Late August is a really good time for vacation as the largest summer heats are slowly going away but we still do not have to worry about the cold encroaching from the November. I would say enjoyable would be a hot day but not too hot and an unjowable day is a super cold or super hot day

Student responses

1. I think this day would have a good temperature to where it wouldn't be too hot or too cold. It also isn't very likely to rain. I would check to make sure the normal range of temperatures on this day is in the mid 70s because I think that would be good for a picnic and to make sure it doesn't normally rain around this time.
2. I would pick this day because most students are done with class early on Friday or don't have it at all. Also, the temperature will be good at this time in September, as it shouldn't be too hot or cold, and the rain shouldn't be bad
3. I would want to hold the picnic when school is in session, so a day in September makes sense. It also tends to be cooler than August and July while still being warm. It is also on my birthday!
4. It is before the July the 4 th holiday, so I assume people will have a good time and also relax during the picnic, even it is a little busy it will be worthy because another holiday is coming
5. It is MY GOAT Messi's birthday! It is summer so the weather would be nice, and we can celebrate great football!
6. it is not hot and sunny

Student responses

1. It's in the summer, which means it should be warm and there shouldn't be too much rain, leading to an enjoyable picnic with nice weather.
2. Its a friday so people wouldnt be too busy, the weather wouldnt be too hot like in earlier weeks or too cold like in later weeks.
3. its my moms birthday and it is towards the end of summer when the weather is warm enough to be outside but not too hot that its unpleasant
4. May doesn't usually have a lot of rain, the temp is warm but not hot.
5. not a week day, not too hot in the middle of summer
6. Not to hot or cold for the end of sept
7. not too hot at the end of may, hopefully no rain
8. People are more likelly to be in town and Friday afternoons are normally less busy for people
9. Right around the start of the summer semester for recruitment. And during the summer so hopefully no rain

Student responses

1. School has started so students will be on campus
2. summer, sunny, hot, no snow or rain, nice for outdoors, past records show its usually sunny and hot
3. The date in May won't be too hot or too cold. Also, I pick weekend so that people should be more available. If it rains, we'll have to change date.
4. The weather is not too hot. School has not started yet. Its during a weekend.
5. This is my frined's birthday
6. This is the date when semester starts and most students are on campus.
7. This will be the day least likely for it to rain.
8. Weather is not burning hot, everyone's back on campus after FDOC, so people have something to talk about
9. y

Student responses

1. 26
2. I chose this day because around this time it isn't too humid or hot and it won't be raining either
3. I feel like late September has the nicest weather
4. I would initially choose a day in late September because the weather is usually warm but not too hot. The picnic would be enjoyable with mild temperatures and no rain, but heavy rain or extreme heat could force us to cancel. I would compare past weather records for different dates to see which one usually has comfortable temperatures and the lowest chance of rain.
5. idk
6. Not too hot, not too cold. Less spring rain.

Student responses

1. I think a Thursday is a great day for a picnic. Enjoyable - sun, cancel rain.
2. In general September should have nice weather and the temperature is amiable for outdoor picnics.
3. It's today, and it's nice weather.
4. Today, have a nice weather
5. Weather is really nice today, plus in general, its better to have a picnic on a cooler day, its more likely to cancel with rain

Student responses

1. I picked this day because not a lot of people have plans or classes on a Friday in the afternoon. If it rains that would be hard, but its not supposed to rain on this day.
2. I think its best because the weather should be best and I would evaluate my choices based on whether there is a lot of rain on that date historically
3. It is after school has started again, and the weather will be warm, but not too hot, and not raining
4. Mid-September is typically cooler and less rainy than both summer and spring months.
5. Saturdays usually are pretty free for people and it should be still pretty warm outside, but not too hot. If it rains, it should not be on that day.

Student responses

1. I chose September 25th as the date for the picnic. I did this because it is now officially fall and would likely not be too warm. Additionally, students do not usually have busy Friday schedules.
2. I chose this because it is a Friday in September. Fridays are always a good choice, September has usually good weather and temps. I think that extreme rain would force us to cancel. Or even extreme heat or cold. I would love a semi cloudy day with no precipitation.
3. I chose this day because during the summer months like June through August the weather might be too hot to sit outside, but in September the weather should still be nice and temperate.
4. In late September, it's usually not too hot or not too cold. My preferred temperature is around 70 to 75 degrees Fahrenheit, and Friday sounds like a good day to have a picnic.

Student responses

1. Near before fall, anticipating lower temps at this time, before it gets too crowded at diag
2. People are free around lunch time on Saturdays, and around that time in August has great sunny weather with little reason to cancel.
3. This is a Friday and during a time when the weather would be nice outside so it would be more desireable to be outside with friends. Could check historical records to check if August 15th would be suitable

Student responses

1. Friday is a good day because it's the end of the week and more people can probably make it. Weather wise, in the beginning of June it is warming up but still not too hot so the weather should be pretty pleasant
2. I think most people don't have classes on Friday and so afternoon time will be free for most people. At June, it is hot but not too hot to make it enjoyable. However, if it was raining I would cancel it
3. This is the friday after most people leave, which would leave the diag mostly depopulated. A less busy event tends to yield higher satisfaction and leaves folks more focused on the picnic.

Student responses

1. I picked June 20th, obviously this is dependent on what the weather is like on that Saturday, but the main reason I picked it was because it wasn't a workday isn't a Sunday for religious reasons. I would use the weather records to evaluate my choice as well
2. I think it is a less rainy month, and Saturdays are days that more people might be free. I would want to check historical records for rainfall and temperature
3. Saturday would maximize availability and I would expect the weather to be not too hot at this time

Student responses

1. Everyone would be excited to come back to school and socialize, there wouldn't be too many conflicts on a Sunday, and the weather is generally still warm but not super hot anymore.
2. The weather is still good at this time but not too hot. It is also on a sunday so not to interfere with a football game.
3. Weather still good, no football game

Student responses

1. I think a sunday would be the best choice, later in september so the weather will be more consistent and a bit cooler than it is now. I would look at precipitation from previous years and high and lows of weather to determine the best date.
2. It will be amazing weather as it starrrts to be fall
3. sunday

We only used 30 years of data, but `n\_years` is 180 in most rows. Explain what went wrong and how you would fix the code.

81 anonymous responses

Using only day component caused error

16

Student notes the code only considered the day-of-month or numeric date component, not month and year, leading to incorrect counts.

Need to filter rows by year range

14

Student says rows should be filtered to include only the desired years (e.g., restrict to the 30-year range) so `n_years` isn't inflated by extra rows.

Ignored month information

12

Student briefly notes that months were not accounted for, implying the grouping should include months.

Counted all qualifying years across dataset

10

Student explains `n_years` is 180 because counting included every year meeting the condition across the dataset, not restricted to the 30-year window.

Response themes for question 2

Grouping misses month component

10

Student states the code ignores month information and suggests grouping by both month and day (or including month in the grouping) to avoid collapsing different months together.

Multiple entries per year inflate count

8

Student points out that several records within the same year (e.g., multiple dates) are being counted separately so n_years reflects number of rows or dates rather than unique years.

Group by month and day to fix

6

Student explicitly recommends grouping by both month and day to prevent mixing dates across months and get correct counts.

Response themes for question 2

Possible incorrect n_years calculation method

5

Student asks whether n_years was computed as a sum of non-missing years or otherwise miscalculated rather than counting unique years.

Student responses

1. Data was grouped by "day" which means the date out of the month, so it's calculating the number of raining day for each day of the month, not each date out of the year.
2. Day was used rather than years?
3. grouping by day groups by the day of the month, not the day of the year since we don't take into account the month each day is in when grouping, leading to many more years of each day than we actually looked at.
4. Grouping by day makes each of the 6 months day of the month all combine into the data for that given day. This is not an accurate representation, and it needs to be grouped by month and day.
5. idk
6. It is combining the 1st day of each of the 6 months per year into one data point.
7. it is grouped by day
8. It is summing together all the dates where precipitation is not equal to zero
9. It only accounted for date, and not month and date.

Student responses

1. Since it only grouped by day and not day and month it did just by the value of the day not actual dates
2. The code is wrong because it doesn't give the year or day correctly
3. The date only range from 1 to 30, no difference in different months.
4. The group only looked at day but not month
5. using only day component
6. You only used the day component
7. You're grouping just by day, not by both day and month.

Student responses

1. .
2. I think that filtering methods is not correct
3. I would create a filter
4. Maybe .groups is wrong. I'm not 100% sure because maybe I just havent seen it.
5. need to filter by row
6. Need to filter rows
7. The units must have gotten mixed up because it was grouping by day only which would mix up the months so we would have to group them separately
8. we conunt years in to different units
9. We didn't filter out the years before 1995 in this command, so it took the data from all 180 years in the table

Student responses

1. We didn't select only the date from 1996 to 2026.
2. We didnt filter out the data before the year 1996 so we are getting 180 years not 30
3. We do not filter the selected month
4. We woulg use 180 years of data or view the datta weri careflu
5. You don't filter it to look at only the last 30 years meaning it will take all 180 years

Student responses

1. check for the days of rainy, instead of summarizing years
2. Currently the years is being calculated by the sum , need to have filtered.
3. Grouping by day causes us to lose monthly information, and the data is aggregated for the 31 days of a month rather than per month
4. It included each date by month as well.
5. it is because it is counting the number of weeks?
6. multiple entries could've inflated
7. only day was used, not seperated by months. would filter by row
8. The 180 comes from the number of days in the range.
9. The group by is dropped which left less rows than we had

Student responses

1. theres multiple entries
2. We didn't account for months.
3. Years is being counted in days in the year instead of labeling the year itself. Not sure where the error is.

Student responses

1. 1
2. Because the data is grouping by day and not years, but then the quote is requiring to summarize by n_years.
3. Because we counted all of the years that there reached that precipitation benchmark of ≥ 1 , so it is listed at 180 for each row
4. don't know
5. I think a large chunk of the data was not sorted correctly
6. It counted the years where it rained excluding the filter
7. It has overcounted by 6 times as much so it is using the wrong units
8. Need to filter more thoroughly by group by
9. No idea

Student responses

1. We extracted from May to October, so if we group by each day, there will be 6 months that are extracted each year. Since we used 30 years, we end up getting $60 \times 3 = 180$ rows of data

Student responses

1. Duplicates?
2. group by month and day, and make sure grouping is correct
3. I'm not sure maybe the fact there were 6 months includede somehow
4. it is counting some months as years
5. It is counting some months as years, we didn't factor in units
6. It's counting some months as years
7. The code did not split the data by month. To fix the code, we could separate the data by month for each day in each year, which would give us n_years of 30
8. There is no consideration by month. As a solution, I would group by both month and days instead.
9. We checked rainy days per year rather than summarizing by year properly, which is fixed across years.

Student responses

1. We grouped by days but it also grouped by month. So we need to group by month and date

Student responses

1. Because we are grouping by day, which for 6 month, $6 \times 30 = 180$
2. Double counting years as it group by month and day and not years
3. I think maybe there were multiple entries per day
4. multiple dates within the same year
5. The error is that the way year was calculated counted the number of days each year
6. The value of `n_years` is based on precipitation values, meaning that each year contributes precipitation, not just 1 year.
7. There is some sort of a mistake with multiplicity
8. We used `groupby` day, which created 'duplicate' years

Student responses

1. Grouped the data by just day, instead of day and month.
2. It uses the wrong unit, so we need to fix the unit.
3. The code grouped only by the day number, so it combined the same numbered day from all six months and created 180 observations. I would fix it by using `group_by(month, day)` so each calendar date is counted separately.
4. The data is grouped by day, but it does not take into account the months. For example, data from the first day in each month is all grouped together, and so on. To fix this, group by month and day.
5. unsure
6. y

Student responses

1. did you set the n_year as the sum of all the years that weren't NA?
2. I think the error is that the sum of the n_years was calculated at an overall 30 years but not for each year?
3. In the code you have n_years as a sum of the prcp which I think is why it gave 180 instead of 30.
4. sum years is not calculating correctly
5. The line where n_years is created is looking at the wrong variable

Each of these dates has 29 recorded temperatures. Why is its mean still `NA`? Explain what went wrong and how to fix it.

82 anonymous responses

Missing data points present

43

Students state the mean is NA because one or more temperature values for those dates are missing within the range.

Exclude or ignore missing values when computing mean

12

Students recommend excluding missing values or configuring the computation to omit NAs so the mean can be calculated.

Data-cleaning/NA-checking mistakes

6

Students point to errors in how NA filtering was written or applied (e.g., forgetting to negate `is.na` or insufficient cleaning).

Other responses

6

Responses the model could not place reliably.

Response themes for question 3

Missing min or max temperature

5

Students specifically identify that either the minimum or maximum temperature is absent, preventing calculation of the daily mean.

Grouping with NA yields NA result

4

A student notes that when the grouping key or aggregated count is NA, the grouped mean computation returns NA.

Units or formatting mismatch

4

Students suggest the NA might stem from differing units or misformatted/incorrectly recorded temperature values rather than simple missingness.

Other responses

2

Student responses

1. At least one record of each of the days grouped by has an NA.
2. because each date contains at least one missing temperature
3. Because for the recorded temperature it's missing there is an NA.
4. Even if one value is NA, the average cannot be computed
5. Every other day has 30 years, so these dates only having 29 could mean that one of the entries was missing an input, leading to a NA value, which would lead to a NA value for the mean.
6. I didn't check if there's any missing data
7. if there's only 29 and not 30 temperatures, it means on one of the years the min or max of that day was not recorded. this would mean that the average of that value would be NA
8. If you have 29 actual counts and one that is NA, then the mean of all of them will be NA. Since $NA + \text{anything} / \text{anything}$ is still NA.
9. In some years there may be data missing causing the entire mean to be NA

Student responses

1. It is NA because we do not have all of the necessary inputs to actually get to the average. As we learned previously if you pass NA through a function the entire function will return NA
2. Maybe an important number or data point for these days is missing and we can't compute the mean if we don't have specific numbers.
3. maybe one of the values on those days is missing and therefore the calculation of the mean fails
4. Missing data points
5. missing temperature is due to instrument damage
6. Missing values were not dropped
7. One of the date temperatures was NA which made the mean calculation NA
8. One of these days must have been missing an entry for tmin or tmax causing the whole average for the day to be NA. So we need to drop those days.
9. one record of these month are recorded as NA

Student responses

1. Our data range is 30 years. So if n_years is 29, that means one of the temperatures for that day is missing, resulting in the whole thing being na
2. since it's missing
3. Some of the data points are missing.
4. The data range is 30 years, but we only see 29 years in the data table
5. The mean is still NA because even if 29 of the 30 years have recorded data, just one value of NA will cause the calculation to result in NA
6. The mean is still NA because mean_temp was calculated from all temp_f's in the mont. If a temp_f is NA for a given day, the mean temp will also be NA.
7. The mean temperature is NA because there are still missing values in the individual temperature values.
8. The temperature was not recorded on one of the days in the 30 year period which causes the computer to average it to NA.
9. Their temperatures were not computed for one of the days in the 30 year stretch, so when they try to average them it returns as NA because one of the values is NA.

Student responses

1. There are 29 recorded temperatures, but our data has 30 years, so one year is missing records for these dates. Use `mean(temp_f, na.rm = TRUE)` to compute the mean without these values.
2. There are days that don't have a temperature (NA) and so when we try to find the average it tries to add the NA to the mean calculation and fails.
3. There are missing data points for one of those days within the time range.
4. There is a missing data value that causes the whole mean to come up as NA, fix it by removing missing values from the table
5. There is missing data
6. There might be missing values on these days, causing NA mean temps since mean of any individual NA value leads to NA overall, a good thing to fix is to drop it
7. There should be 30 recorded temperatures for each of these dates instead of 29, meaning that one is N/A. We can remove the one for each of these dates that is N/A and then take the average of the new data subset.
8. There should be 30 recorded temperatures. This means that one of the `temp_f` is NA, making the mean NA. Should remove NA values.

Student responses

1. there was 1 missing recorded day, it makes the mean function NA too
2. There was one missing data point which is NA. That means the mean can't be calculated because it will show the answer as unknown
3. This is because if we have one NA then if we do mean, it gives NA as a result
4. This most likely that there was a missing data place in either tmax tmin or both and because of this when you caculated the average for that day it resulted in a NA. Note that whenever you do any operation with NA it will always result in NA so this is mostlikely what had happened. To fix it, you would need to add logic in the mean to add on days where it was not NA
5. We only have the mean of the 29 responses from a specific date, not all 30 years.
6. We should have 30 recorded temps, since there is one NA value for each date the mean() returns NA
7. what went wrong was the tmax or tmins were missing, leaving NA that makes it unknowable.
8. While calculating the mean, you did not check for any NA values, so when it calculated the mean, if any day was missing for whatever reason than it can not calculate the mean and is left as NA.

Student responses

1. Because there is one NA temperature in the dataset, so the mean will give NA. To fix it, use na.rm
2. even if there's one na, the entire mean will be na because we don't know what it is, instead when calculating the mean we can remove na
3. i think there were missing data points, and it didn't exclude those when calculating the mean
4. If contain a NA, the mean would be NA because it's unknown. We need to add "na.rm=TRUE" in the code
5. In the 30 year period, one of the years had NA for either tmin or tmax, causing the average temp to be NA, causing mean temp to be na. You can fix this by doing na.rm = TRUE which will ignore the na since it's just 1 value.
6. Its because there was a value that was NA for one year. You can add the argument that ignores NA values to fix it
7. maybe we do not filter the NA raw value
8. One NA kills the whole mean. The solution is to remove it and take the mean of the 29 recorded temperatures.
9. one temperature is missing out of the 30. we need to tell it to ignore all missing values and only account for the data we have

Student responses

1. R does not assume or drop a number for NA value. So we would have to manually decide weather to drop it or put in a value so R can compute it
2. The mean is still NA because mean() returns NA if even one value is missing. Adding na.rm = TRUE inside the function, like mean(temp_f, na.rm = TRUE), will calculate the average using the 29 recorded temperatures.
3. there are missing inputs. have r exclude it when there is no value

Student responses

1. access out of scope?
2. Maybe the NA was not cleaning from the tmin and tmax, so the functions involve NA will give NA.
3. Sometime must have happened where there as a NA for one day out of the 30 years and one NA will cuase the avergae to be NA as well. So we should drop an NAs
4. the mean is still na becuase there is no ? before the is.na
5. the na is not cleaning the data enough
6. Use is.na but forget a !

Student responses

1. Because for mean() function, it doesn't really filter out those with NA values. Once an NA exists, then the result of the mean would be NA.
2. If any numbers are missing, you can't take the mean of it. You need to get rid of those dates.
3. is the temp still in two different measures (celsius/fahrenheit)?
4. Maybe was recording in wrong degree?
5. There are no different data points
6. These temperatures that show up as NA were recorded in Celcius

Student responses

1. Either the max temp or min temp is missing from this day. You will need to exclude these days from the final data set
2. Either the min or max might have been missing, therefore there was no way to calculate those days' mean temperature.
3. it is missing a max or min temp for those days so the mean returns NA
4. The mean is NA because one of the min or max temperatures in the original data set is NA due to a lack of data, causing the average to be NA.
5. They don't have a tmin or tmax, so R couldn't average them. You should go find those values

Student responses

1. a missing value in group makes mean return NA
2. It still has NA responses because at some points in time the data collection tower did not collect data for one reason or another. To fix this, we'll use true and false data while only keeping dates that have a recorded temp.
3. The mean is NA because the formula may miss.
4. There must be some missing data points present in the NA so that makes it impossible to calculate the mean since not all the entries are fill

Student responses

1. because of NA
2. I think that the mean temp could be missing from one data point or recorded differently
3. Some of them are missing because not all months have 30 days. In this case it can range from 28-31
4. The mean is still "NA" because it is trying to average over all the values, regardless of weather or not a value is "NA".

Look at the dates at the top of this ranking. Did we find averages closest to 72°F? Explain what went wrong and how you would fix it.

80 anonymous responses

Use absolute differences

20

Points out distances can be negative and that one should take absolute values to measure closeness.

Outlier or wrong sort direction

15

Suggests an outlier or that the sorting direction (ascending vs descending) is incorrect and causes wrong ordering.

Brief yes/no or negation

10

Very short confirmation or denial without explanation.

Sign error in subtraction

10

Notes subtracting 72 from the mean yields signed values that emphasize farthest rather than nearest; implies taking absolute value or otherwise using magnitude would fix it.

Response themes for question 4

Empty or unclear response	8
No meaningful reason given; response is just a placeholder or punctuation.	
Temperature units misunderstood	7
Claims the temperatures are in a different unit (Celsius), implying the comparison to 72°F is invalid.	
Wrong sorting key used	6
Indicates the code arranged by the wrong parameter; should sort by the distance from 72 (closest to zero) rather than the raw difference or another field.	
Computation formula error	4
States the calculation formula itself was incorrect, causing wrong results.	

Student responses

1. Absolute differences
2. It is arranged from lowest temperature to highest temperature, we should subtract from 72 and get the absolute value if we want absolute value
3. Its sorted the wrong way, as it contains negative values we need to look at abs value
4. No, since we did temp - 72, some of the values are negative (and therefore lower) but we meant to calculate the distance from 72, so we should use absolute value in our calculation
5. No, the absolute difference must be used.
6. No, the code ranked the most negative differences first, so it selected the coldest dates instead of those closest to 72°F. I would calculate the distance using `abs(mean_temp - 72)` and then sort from smallest to largest.
7. No, you need to use absolute differences
8. No. These are negative values. We should use abs
9. No. We should attach an extra absolute value.

Student responses

1. The chart is arranged by smallest value to largest, so negative values appear before any values closer to 0. To fix this, we can take the absolute value of the mean temp - 72
2. the distance is both negative and positive, so the furthest negative seems to be the closest, when it is actually the lowest away. Need to take absolute value of it
3. The lowest distances are negative, you need to take absolute value
4. This is recording the values furthest colder than 72 degrees because it sorts by lowest and so it starts with the lowest negative values.
5. Those dates are the coldest days since they are less than 72 degrees F, need to do absolute value
6. We are sorting by distance wrong, we need to take the absolute value of distance
7. we did not find it, should use absolute diff
8. We didn't look for the lowest absolute distance (distance from 0), so the top of the ranking is numbers that are way under 72 degrees F. We should arrange by absolute value.
9. we found the date with the lowest average temperature, to fix this we can arrange by the absolute value of distance to find the date with the average temperature closest to

Student responses

1. We found the temperatures farthest away in the negatives from 72. We have to take the absolute value of the distance
2. You subtracted 72 from the mean temperature instead of the other way around

Student responses

1. Arrange arranges from lowest to highest, so instead of getting the temperatures closest to 0, we ended up getting the temperatures that were the furthest away from 72 that were below 72
2. It's now from the farrest to the closest, we need to add `arrange(desc(distance))`
3. no, signage issue
4. No, we need to arrange it from descending.
5. Reverse the sort order
6. sorting in the wrong direction
7. The `arrange()` function sorted by most distant to least distant, rather than least distant to most distant as intended. The table shows the days with the temperatures most distant from 72 on top.
8. The current `arrange` call puts negative values first. A descending `arrange` should be used instead to get the dates with the lowest positive dfference
9. The distances are sorted in ascending order instead of descending order. We have to add in the descending command to fix it.

Student responses

1. we arranged by distance before checking absolute value so it starts with smallest/most negative values, which are the coldest temperature days in our daterange.
2. we arranged by distance, but its giving us the farthest distance first. we just have to arrange it in the other direction
3. We didnt... I think we found all of the lowest mean temps. I would maybe arrange it by smallest distance?
4. we found the averages farthest from 72 degrees fahrenheit because when we used arrange, we inputted distance which would give us the results that are the farthest from out desired value.
5. We found the averages furthest from 72. We need to use the `arrange(desc())` function.
6. Wrong order asc/desc.

Student responses

1. did not find averages closest to 72F
2. is does not include the other data
3. must have used absolute distance from 72 f
4. No we didn't find average close to 72F
5. NO we didn't find data closest to 72
6. No, responses are not all present so we need to clean the data?
7. No, there are around 40s.
8. No, we didnt find averages closest to 72, im not exactly sure what happened, maybe a list was sorted wrong.
9. we should sort by absolute value

Student responses

1. Yes

Student responses

1. arrange orders in ascending order, so these are actually the values that are furthest away from 72, so we need to use desc() to get it in the correct order.
2. Distance is bugged. We need to convert all neg values to positive
3. mean_temp - 72 calculates the values furtherest from 72, not the closest.
4. no this didn't end up working, i'm not sure how to fix this though
5. No. The temperatuers on the top rows were around 47F, which is significantly lower than the average of 72F. The arrangment was done by the distance not the temperature.
6. problem is at that we define distance = mean_temp - 72
7. Ranked by distance, which was mean temp - 72. So if you're trying to see dates with mean temp close to 72, distance should be 0
8. The distances are in order of furthest from 72 degrees at the top
9. There was an error with the sign

Student responses

1. those dates are the most coldest days; since they are smaller than 72. we want to find the smallest absolute value from 0

Student responses

1. .
2. it put the coldest days on the top of the list because subtracting all values by 72 makes all temperatures go down. We want closest to 0.
3. na
4. No we didnt, we found the days furtherest from 72, so we have to sort it in ascending order
5. No, arrange returned the days the most far from 72.
6. We found averages furthest from 72. We need to account for the positive versus negative difference between the value and the mean and use desc()
7. We found the dates farthest from 72 F, arrange sorts from lowest to highest, so most negative numbers come first, fix this by adding - before distance in the arrange function
8. We need to put a minus sign next to distance so we can arrange it in the opposite order.

Student responses

1. No distance on the top is higher
2. no, we ignore units
3. No. The averages we found do not have a mean closest to 72 degrees. I think what happened is that it is giving me a listing based off of distance which is not what I want. That tells me how far I am away
4. the temperatures are in celsius
5. There was an issue with conversion from celius to fahrenheit
6. We found averages that weren't in the right unit
7. wrong of the catagorising

Student responses

1. I think it ranked the dates by the furthest temperature distance from 72 degrees F instead of closest. We would just need to arrange by minimum distance.
2. Im not too sure on what went wrong but I wonder if it had to do with compariing the wrong tempature intervals. Like maybe we are comparing an average and not a general day or maybe vice versa?
3. It is sorting the distance from 72 not the temp
4. no, we need to calculate the distance from 72 and sort by that instead
5. We need to sort by smallest absolute value, not smallest value.
6. wrong parameter for arrange, it should be how close distance is to 0, could compute abs value of distance and go downwards

Student responses

1. It should compare the distance. / remove outliers
2. no, have some mistake in average
3. Our averages were not close to 72 degrees, using absolute difference
4. The formula for calculation went wrong

The rain rule can choose a cold date; the temperature rule can choose a wet one. Both rules only look at averages and ignore variability. Propose a rule that addresses more than one concern for this picnic. What would you calculate for each observation, and what would you summarize across years for each calendar date? Does your rule make any tradeoffs?

82 anonymous responses

Combine weather metrics into a score

25

Proposes computing a combined desirability score per observation that blends precipitation and temperature (e.g., sum, weighted sum, or distance from ideal), and then summarizing that score across years for each calendar date (mean or median). Acknowledges tradeoffs based on weighting between rain--

Other responses

14

Responses the model could not place reliably.

Response themes for question 5

Include variability in rule

10

Advocates adding measures of variability (e.g., standard deviation) for temperature and/or precipitation per date across years and using those alongside means to avoid choosing dates with high variability. Notes this adds complexity and may trade off average desirability for stability.

Prioritize one metric then tiebreak

9

Suggests a hierarchical rule that first selects by one criterion (least precipitation or below threshold) and then breaks ties using the temperature closeness to a target value; implicitly summarizes each metric across years for dates to compare. Notes tradeoffs by privileging one factor over the其他

Response themes for question 5

Non-weather scheduling concerns

8

Raises availability and scheduling conflicts as selection factors rather than specific weather summaries; implies different data to compute (calendar availability) and different tradeoffs unrelated to weather optimization.

Simple threshold plus proximity rule

7

Gives a concise numerical rule: require precipitation below a threshold and choose the date with temperature closest to a target (per-observation precipitation and temp; summarize across years with frequency of meeting threshold and average/median temperature). Mentions tradeoffs of strict filtering

Response themes for question 5

Use temperature range constraints

5

Specifies selecting dates whose observed temperatures fall within a preferred range and precipitation below a limit, summarized by proportion of observations meeting the range and typical bounds across years.

Combine rules with frequency threshold

4

Proposes combining criteria such that a date must meet frequency-based conditions (e.g., less than X% rainy days) and be closest to an ideal temperature when summarized across years.

Student responses

1. A rule that could be implemented is to create a new column in our data titled good_picnic where the values would be true only if it is not a rainy day and the temperature is not cold. We need to essentially combine the two tests.
2. calculate temperature and distance variable filtering by both and summarize avg considering both trend across years.
3. Choose a day that combines both the rain rule and the temperature one. A day that has less than 1 mm or rain and is above 65 degrees F. Do this by averaging over the last 30 years
4. Choose days with least amount of historical rainfall (rain average below a certain threshold). Then out of those days, choose day with average temperature closest to 72 degrees F. A potential tradeoff could be that it could choose a day with some rainfall and some temperature variability.

Student responses

1. For each calendar date, I would incorporate two rules to account for weather. Temp has to be closest to 72 degrees and rain has to be closest to 1 mm. This incorporates both the temp rule and rain rule, as this would pick the average day that had the temp closest to 72 degrees and closest to 1 mm of rain, by weighing both of these factors equally
2. I think I would make a formula that somehow calculates both factors and come up with a index that I can compare.
3. I think what we can do is make a rule that incorporates both of the previous years. What if this rule takes into account rainy years and also the temperature, so essentially if we calcualte a general average between both statistics and the lowest score is the one that moves on. Not completely sure how to make that average so that its fair but maybe the most middle. I would say the tradeoff however is that you are probably never getting the best of either choice but rather the average change both

Student responses

1. I would be concerned about proximity to a rainy day. If it was raining the day before, the ground still might be wet. To calculate this, I would make a rule that a good day for a picnic cannot be within one day of a rainy day and filter out any days that do not follow this rule.
2. I would calculate the average distance of temp_f to 72F and the sum of rainy days of every day. Then, compute temp_distance * rainy_days and pick the lowest
3. I would calculate the average for each observations and i would summarize the data across the years for each claendar. not waware of any tradeoffs
4. I would create a score for both temperature and precipitation and see which one has both for each observation. This is what would be summarized, yet this does make a tradeoff of one factor potentially outranking the other.
5. I would filter by dates that rain of 0 and temperatures that are between a certain range, like 72 to 80 F. You could try to find a way to give a weight to every rain and rank it with a score and the same thing with temp and average those together

Student responses

1. I would look for the days that have the smallest amount of rain while still being a certain temperature, as well as cloud cover if possible. For the smallest amount of rain, I would find the average rain across the years, while the temperature would be the same thing we did before. For the cloud cover I would assign a boolean value to a standard, and then average that out. It doesn't consider the day, meaning that it could happen on a Tuesday, which would be inconvenient.
2. I would take into account both rain and temperature. calculate the rankings for each and assign them a number with the number being their ranking, like best day for temp is 1 and same for the rain (low amount of rain gets 1), then for each day combine those numbers and then rank by lowest number, tie breaker being their weather rank

Student responses

1. I would use the similar method. However, I will add a weight score respectively for precipitation and temperature. Those who are most dry or closest to 72F would be endowed with a higher weight score. Then I will add up those two weight scores and determine the date with highest combined weight scores.
2. make a function have are linear combination of the rain rule and the temperature rule
3. Make/learn an objective function that is some linear combination of rain and temperature weighted by how much you care about either temperature/rain. Then minimize/maximize. For our variables, we could do average temp per day and total precipitation, taking medians across years as our "averaging" scheme.
4. Maybe a rule can be a combination of the rain rule and the temperature rule. To calculate this observation I would probably try to set weights or create some type of score. This will summarize both temp and rainy days. The tradeoffs for this is that it might weight the temp and rain differently

Student responses

1. My rule would choose days that must have the least amount of precipitation while still being relatively close to 72 degrees. Rain should be less than one mm, and temperature can fluctuate 5 degrees around 72.
2. My two main rules would be zero rain, or as close to zero as possible. My preferred outdoor activity temperature is 75 degrees, so as close to that as well. Averaging out the two would be smart, but rain is way more important so that will factor in more than the other.
3. Take a sum of each variables ranking, and the lowest sum wins.
4. Take the ranking of each variable and add them together, the then sort by who had the best combined ranking
5. temp difference/ 72 + rainy days / no of days in the months and sleect smallest number, not sur ehov we ware wieghting both factors, not standardised or equal
6. We can assign values to coldness and wetness on a scale of 1-5 and sum both scales
7. you can create a score that is the sum of the desiribility of the two inputs.

Student responses

1. Compare the both outcomes, find the most fit day from top of each outcome, allow the matching range to be within 3 days.
2. consider aesthetic rating of each day/ month
3. For each observation, I would calculate the average temperature and mark whether the day had less than 1 mm of rain and a temperature between 65°F and 80°F. For each calendar date, I would calculate the percentage of years that met both conditions and choose the date with the highest percentage. The tradeoff is that this rule may exclude some slightly rainy or cooler days that could still be enjoyable.
4. I will propose a hurricane rule where it shows the likelihood the hurricane will happen in a specific date. By summarizing the time we will get a hurricane, we can avoid that day. There exists trade off at missing the other key factors.
5. I would also consider other variables like sunlight, because some may want sunlight but some may not.

Student responses

1. Is the day during the fall? Some people prefer fall picnics to summer as the leaves are a different color. A simple bool expression can be calculated for each date by comparing it to when fall starts. The tradeoffs are that fall days are cooler.
2. Look at past extreme rare weather conditions, including when the day is too hot to go outside although it didn't rain. I will need to calculate the max temp.
3. Maybe commute a new sheet and then compare wet and temp. choose the day in 72F and not raining
4. Maybe make some sort of matrix, with each date with a suitable rating of the :suitability: for picnics. Then we can more easily observe the good picnic days while accounting different factors. Also what if we design a rule that it has to be a famous celebrity's birthday, so no matter how cold/wet/unpleasant the weather is, people can always have some fun!

Student responses

1. My rule could be a combination of both the heat and weather rules, or perhaps a combination of heat and wind as a rule since those are both essential factors, we can calculate for both an ideal wind speed and an ideal temperature, and we would summarize a combination metric with closeness to the parameter set as ideal for wind speed + closeness of temperature, does make some tradeoffs like feels like, humidity, etc
2. One rule could be the UV index outside to determine how much sunscreen is necessary or the amount of foliage and fall colors present outside.
3. One rule that might cover more than one concern would be overall intensity of sunlight throughout the day. Shorter days in the winter will have less sunlight (still colder), and cloudier, rainier days will also have less sunlight (even during the warmer summer months). The tradeoff of this is that this kind of data might not be as readily available.

Student responses

1. Rule: What is the best date for this picnic if I want the picnic to occur on a hot, summer day with no rain. I would have to define what hot means in degrees and provide a range of dates. I would also have to filter out dates that on average have rain. I think the tradeoff would just be what if the picnic should happen in the fall?
2. The sun rule, factoring the amount of sunlight for a day, which can factor in temperature and precipitation. I would calculate average UV, time of sunshine, and average temperatures so that rainfall and potential cloudy days are already factored into this decision.

Student responses

1. A rule would ensure that the dates are not too variable, limiting variance and ensuring trim
2. Also look at deviation from the mean to choose the least risky date
3. balance temperature, rain, and variability. choose dates with consistent mild weather. prefer dates with low weather extremes
4. I think I will create another column that calculates the variability; also, some students may prefer cold days and wet days, so can let them complete a survey and then pick the one that most people agree with
5. I think looking entirely at temp and precipitation variability after finding out the best temperature days would be a good approach. After finding the top 30 best temperature days, we should sort based on the lowest variability in those day's temperatures and precipitation measurements. This will make it less likely that we cancel the event.
6. I would calculate a std deviation of 5yrs and 5 degrees and then look to see if we have any dates that are within that deviation for both rules above.
7. include standard deviation threshold for both criteria

Student responses

1. It would be better to use the max temperature because chances are that the picnic would be during the peak temperature of the day. You should also examine the standard deviation of both the temperature and rainfall. You would want something with std dev. with 3 degrees
2. Then we could look at days that tend to vary the least
3. To address more than one concern, a rule could be to have a certain range of precipitation and range of temperatures that both have to be true in order for the date to be considered good for the picnic. Instead of only looking at averages, it would consider the variation of the values as well.

Student responses

1. A rule could be to pick the week that historically has the least rain, and then pick the day with the best average weather. For each calendar date I would summarize average precipitation and average temp over the years. This rule has the tradeoff of placing rain over temperature in terms of importance, but I believe that a lack of rain is more important than the most ideal temperature for a picnic.
2. combine both of the rules and find a day that has less than 50% of the days rainy and is closest to an ideal temperature
3. Finds the days that are both closest to 72F and average less than 1mm of rain.
4. I would look at the 30 closest to 72 degrees, then pick the one with the lowest rain days, since temperature is probably more stable across years than rain. I would also prefer a weekend day, if the resulting day wasn't a weekend I would pick the closest weekend day.

Student responses

1. I would look for a date that is rainy less than half of the last 30 years, and one that is on average at most 5 degrees off of 72 F. This trades off the extremely dry or fair weather days that fail the other metric, even if this year they happen to be the best choice
2. I would prioritize the temperature rather than the rain since the rain is more random and does not really have much to do with the actual day. We do know that the middle of the summer will be hotter and may and october will be a little cooler. So pick the best temp
3. Maybe give the ranking for both rule orderings, and add the two rankings for different days together, find the smallest sum. It's the best date combining both rain and cold rule.
4. pick least rainy and then temp closest to 72
5. To pick the best day I would want to pick a day that has the lowest chance of rain while still having decent temperatures, so I would probably pick the day whose average temperature is at least 60 and has the least number of rainy days.

Student responses

1. a rule could be the most non crowded day
2. calculate the amount of people going out,
3. filter some days in summer, and select some sunny days
4. I think day of the week it would be on this year is also really important, though I'm not sure exactly how we'd write a pipeline for that. A rule I'm thinking of otherwise is average max and average min temperatures?
5. I want to make a rule taht the date should be a day that students are willing to spend the most time out, for example, 5 hours will be better than 3 hours. I would calculate the time and try to find a most popular week then the most popular day. my rule might make tradeoffs because the results can be really colse.
6. I would calculate temp and precipitation and combine them into one rule. I would do the daily max temperature and total precipitation, and make sure there's no rain and not too hot
7. I would combine weather metrics into days that are thursdays or fridays, balancing weather with which days of the week
8. look at precipitation data such as multi year mean and frequency of the dry days

Student responses

1. For each observation I would make sure there was less than one mm of rain, and filter the weather in a range of 69 to 80 F. Across calendar years I would do this by summarizing mean temp and taking out all precipitation. But removing all precipitation hides variability—summarize mean and also variability (e.g., percent dry days and temp spread). I would be clearer about per-observation checks versus across-year summaries and specify exact statistics to report.
2. for me, I'd say a rainy day would be greater than 4 mm of rain and a maximum temp of about 74. I say this because using a maximum temp finds us the temp we're likely going to have during the day instead of averaging the daytime and nighttime temperature values. Then I'd combine both rules to find which days are most likely to have these in common. The rule does risk the temp not being totally accurate or a burst of rainfall in the middle of the day, but I think it's likely to select a good day.
3. I think a good rule would be that the rain needs to be under 1mm and the temperature needs to be with 8 points of 72 degrees

Student responses

1. My rule would be less than 0.1 inches of average rain and any temp within 5 degrees of 72. The tradeoff would be that no day would be the perfect fit likely.
2. Needs a temperature between 60 and 79 degrees and less than one mm of rain. The rule can make tradeoffs by potentially going out of the range for one to secure the other
3. Pick a date where the min temperature is above 68, and the max temperature is below 80, and the precipitation is below 1 mm.
4. prcp is below one and the temp is closest to 65

Student responses

1. find the date with a min and max temperature range that is closest to the threshold temp range of 68-75. average min and max temperatures for each day across the years to find the average range and then find the distance from the threshold for each average.
2. I think that one thing I would do is find a temperature closer to 65 degrees, so that it somewhat accounts for humidity (and therefore rainy weather), but is still nice outside for a picnic.
3. I would honestly start only looking at months that you are confident won't be too cold or too hot, so maybe June and September. Then from there only look at rain and look at it a lot deeper. This way we rule out the temperature factor and only focus on the rain.
4. I would look for a weekend day with no rain and a temperature that is between 67 and 77. There may not be a perfect day but we would manually pick the closest one.
5. One rule can be that we can only choose a date in June, July, or August. Those days are guaranteed to be warm and they have a low likelihood of being rainy.

Student responses

1. If there is historical record of the UV index for each day of the past 30 years, finding days where the index is closest to around 6-7.
2. Maybe set a threshold for each one, where for the temperature, if the temperature reaches a minimum temp, then we can choose that and then cancel it if it is wet on that day. Then you can take the average of the entries and summarize across the years for each calendar date and choose the best date for the picnic. There would be some tradeoffs with people who might enjoy rainier weather?
3. The humidity rule could help decide if it is wet and warm outside, which could help rule out both a cold date and a wet data. I would calculate the mean humidity and sort by day and month. It does have tradeoffs because we are not getting the exact temperature which could be important
4. what about rain temperature rule

- Define a weather-based score for each day. - Group the same calendar date across years, using month and day. - Summarize each date while handling missing values. - Rank the calendar dates. - In an R comment, report a recommended date and supporting value.

82 anonymous responses

Other responses

82

Student responses

1.

```
# event_days |> temperature_ranking <- event_days |> mutate(temp_f = ((tmin + tmax) / 2) * 9 / 5 + 32) |> group_by(month, day) |> summarize( n_years =  
sum(!is.na(temp_f)), mean_temp = mean(temp_f, na.rm = TRUE), .groups = "drop" ) |> mutate(distance = abs(mean_temp - 72)) |> arrange(distance)  
rain_ranking <- temperature_ranking |> mutate(rainy = prcp >= 1) |> group_by(month, day) |> summarize( n_years = sum(!is.na(prcp)), n_rainy = sum(rainy,  
na.rm = TRUE), .groups = "drop" ) |> arrange(n_rainy) final_ranking <- rain_ranking |> mutate(final_rank = distance + n_rainy)|> arrange(final_rank) #Ranking  
the score arrange(-final_rank) # Recommendation: 10-8-2026 is the best due to weighted distance and rain
```

Student responses

1. # Recommendation: ### Define days that have least amount of rain (added across month and date) ### Out of the top 10 least rainy days, choose one with best average tempdir(least_rainy_days <- event_days |> mutate(avg_temp = (tmin + tmax) / 2) |> group_by(month, day) |> summarize(total_prdp = sum(prcp), avg_temp_cross_years = mean(avg_temp), .groups = "drop") |> arrange(total_prdp) |> head(10) least_rainy_days |> arrange(desc(avg_temp_cross_years))
2. best_weather_score <- event_days |> mutate(temp_score = as.integer((tmin>12) & (tmax<30))) |> group_by(month, day) |> summarize(mean(temp_score)) # Recommendation: best_weather_score
3. days_score <- event_days |> mutate(good_temp = as.integer(tmin > 18 & tmax < 26), no_rain = as.integer(prcp < 0.5)) |> group_by(month, day) |> count(good_temp, no_rain) days_score # Recommendation:
4. event_days |> # Recommendation: mutate(rain = as.integer(prcp < 1))
5. event_days |> mutate(score = rain_ranking) head(event_days) # Recommendation:

Student responses

1. `event_days |> mutate( avg_temperature = (tmin + tmax) / 2, score = -abs(avg_temperature - 20) - prcp ) |> group_by(month, day) |> summarize( avg_score = mean(score, na.rm = TRUE), .groups = "drop" ) |> arrange(desc(avg_score))` # Recommendation: Choose the date in the first row because it has the highest average weather score. # Recommendation:
2. `event_days |> mutate( temp_f = ((tmin + tmax) / 2) * 9 / 5 + 32, good_day = prcp < 1 & temp_f >= 65 & temp_f <= 80 ) |> group_by(month, day) |> summarize( n_years = sum(!is.na(good_day)), percent_good = mean(good_day, na.rm = TRUE) * 100, .groups = "drop" ) |> arrange(desc(percent_good)) |> head() #` Recommendation: Choose the first date shown because it had the highest percentage of dry days between 65°F and 80°F.

Student responses

1. `event_days |> mutate(weather_score = (tmax - tmin) - prcp) |> group_by(month, day) |> summarise(avg_score = mean(weather_score, na.rm = TRUE),
 .groups = "drop") |> arrange(desc(avg_score))` # Recommendation: Choose the calendar date with the highest average weather score (avg_score).
2. `event_days |> mutate(score = as.integer((tmin > 12) & (tmax < 30))) |> group_by(month, day) |> summarize(m = mean(score)) |> arrange(m)`
3. `event_days |> mutate(score = as.integer( (tmin > 20) & (tmax < 29))) |> group_by(month,day) |> summarize(mean(score))` # Recommendation:
4. `event_days |> mutate(score = as.integer( (tmin > 22) & (tmax < 30))) |> group_by(month, day) |> summarize(m = mean(score, na.rm = T)) |> arrange(m)` # Recommendation:
5. `event_days |> mutate(score = as.integer( (tmin > 22) & (tmax < 30))) |> filter(month == 7) group_by_(month, day) |> summarize(mean(score)) |> arrange(m)` # Recommendation:

Student responses

1. `event_days |> mutate(temp_c = (tmin + tmax) / 2, temp_f = temp_c * 9 / 5 + 32) |> mutate(rainy = prcp >= 1) |> group_by(month, day) |> summarize(n_years_rain = sum(!is.na(prcp)), n_rainy = sum(rainy, na.rm = TRUE), .groups = "drop", n_years_temp = sum(!is.na(temp_f)), mean_temp = mean(temp_f), dist = abs(mean_temp - 72), score = ((n_rainy > 10) * 100) + dist) |> arrange(score) # Recommendation:`
2. `event_days |> mutate(score = as.integer( (tmin>12) & (tmax<30) ) ) |> group_by_(month,day)|> summarize(m = mean(score))|> arrange(desc(m)) # Recommendation:`
3. `event_days |> mutate(score = as.integer((tmin > 12) & (tmax < 30))) |> group_by(month, day) |> summarize(mean(score), na.rm = T) |> arrange(desc(m)) # Recommendation:`

Student responses

1. `event_days |> mutate(score = as.integer((tmin > 22) & (tmax < 30))) head()` `temperature_ranking <- event_days |> mutate(temp_f = ((tmin + tmax) / 2) * 9 / 5 + 32) |> group_by(month, day) |> summarize( n_years = sum(!is.na(temp_f)), mean_temp = mean(temp_f, na.rm = TRUE), .groups = "drop" ) |> mutate(distance = abs(mean_temp - 72)) |> arrange(distance) # Recommendation:`
2. `event_days |> mutate(temp_f = ((tmin + tmax) / 2) * 9 / 5 + 32)|> mutate(score = as.integer(80 > temp_f & temp_f > 70 & prcp == 0)) event_days`
3. `event_days |> mutate(score = as.integer( (tmin > 12) & (tmax < 30)))|> group_by(month, day) |> summarize(m = mean(score), .groups = "drop") |> arrange(desc(m)) event_days # Recommendation:`
4. `event_days |> group_by(month, day) |> Filter(mean_temp > 72 , mean_temp < 72)`

Student responses

1. `event_days |> mutate(rainy = prcp >= 1, average = rainy * 2 + 2 * ((tmin + tmax) / 2) * 9 / 5 + 32) |> group_by(month, day) |> summarize( n_years = sum(!is.na(prcp)), n_average = sum(average, na.rm = TRUE), .groups = "drop" ) |> arrange(n_average) # Recommendation: August 31st`
2. `event_days |> mutate(score = as.integer( (tmin > 12) & (tmax < 30))) |> group_by(month, day) |> summarize(m = mean(score, na.rm = T)) |> arrange(desc(m))`
3. `event_days |> mutate(temp_f = ((tmin + tmax) / 2) * 9 / 5 + 32) |> group_by(month, day) |> summarize( n_years = sum(!is.na(temp_f)), mean_temp = mean(temp_f, na.rm = TRUE), .groups = "drop" ) |> mutate(distance = abs(mean_temp - 75)) |> arrange(desc(distance)) #Recommendation:`
4. `event_days |> mutate(score = as.integer( (tmin > 12) & (tmax<30))) |> group_by(month, day) |> summarize(m= mean(score)) |> arrange(m)`
5. `event_days |> mutate(score = as.integer( (tmin > 22) & (tmax < 30)))|> group_by_(month, day) |> summarise_(mean(score)) ) # Recommendation:`

Student responses

1. `event_days |> mutate( score = as.integer((tmin > 22) & (tmax < 30)) )> group_by_(month, day) |> summarize(m = mean(score)) |> arrange(desc(m))`
2. `event_days |> mutate(avg_temp = (((tmin + tmax) / 2) * 9 / 5) + 32) |> group_by(month, day) |> summarize( avg_temp = mean(avg_temp, na.rm = TRUE), prcp = mean(prcp, na.rm = T), .groups = "drop") |> filter(avg_temp >= 70) |> arrange(prcp) # Recommendation:`
3. `event_days |> mutate(score = (tmin > 12) & (tmax < 30)) ) |> group_by(month,day)|> summarize(m = mean(score, na.rm = T)) |> arrange_(desc(m))`
4. `event_days |> mutate(score = as.integer( (tmin > 12) & (tmax < 30)) ) |> group_by(month, day) |> summarize(m = mean(score)) |> arrange(m)`
5. `event_days |> mutate(score = as.integer( (tmin > 12) & (tmax < 30)) ) |> group_by(month, day) |> summarize(m = mean(score)) |> arrange(m) #`
Recommendation: I would set a weight on the temp and rain and create a score
6. `event_days |> mutate(score = asinteger( (tmin > 12) & tmax < 30))|> group_by(month, day) |> summarize(m = mean(score)) |> arrange(desc(m)) #`
Recommendation:

Student responses

1. `event_days |> mutate(score = prcp > 0.2, tmin>20, tmax<30)|> group_by(month,day)`
2. `event_days |> mutate(temp_nice = ((tmin >= 12) & (tmax <= 26))) |> mutate(norain = (prcp == 0)) |> group_by_(month, day) |> summarise( n_years = sum(!is.na(prcp)), #Recommendation: Choose a date with a temperature between 68 and 80, and no precipitation.`
3. `event_days |> mutate(temp_score = tmax >= 21 & tmax <= 27) |> group_by(month, day) |> mutate(rain_score = prcp == 0) slice_sample(event_days, n=10) # Recommendation:`
4. `event_days |> # head() # weather-based score: mutate(tavg = (tmin + tmax)/2, na.rm=T) |> group_by(month, day) |> mutate(score = 0.1 * tavg + prcp) |> summarize(n_years = sum(!is.na(score)), median_score = median(score, na.rm = TRUE), .groups = 'drop') |> arrange(desc(median_score)) # Recommendation: # July 22`
5. `event_days |> head() # Recommendation:`
6. `event_days |> head() # Recommendation:`
7. `event_days |> head() # Recommendation:`
8. `event_days |> head() # Recommendation:`
9. `event_days |> head() # Recommendation:`

Student responses

1. `event_days |> head()` # Recommendation:
2. `event_days |> head()` # Recommendation:
3. `event_days |> head()` # Recommendation:
4. `event_days |> head()` # Recommendation: `# mutate( score = tmax - prcp ) |> group_by(month, day) |> summarize( avg_score = mean(score, na.rm = TRUE), .groups = "drop" ) |> arrange(desc(avg_score))`
5. `event_days |> head()` # Recommendation: `event_days |> mutate(weather_score = as.integer((tmin > 22) & (t_max < 30))) |> group_by(month, day) |> summarize(m = mean(weather_score, na.rm = T) |> arrange(m)`
6. `event_days |> head()` # Recommendation: `event_days |> mutate(temp_f = ((tmin + tmax) / 2) * 9 / 5 + 32) |> group_by(month, day) |> summarize(n_years = sum(!is.na(temp_f)), mean_temp = mean(temp_f, na.rm = TRUE), mean_prcp = mean(prcp, na.rm = TRUE), .groups = "drop") |> mutate(distance = abs(mean_temp - 72), score = mean_prcp * distance) |> arrange(score) # choose June 26`

Student responses

1. `event_days |> head()` # Recommendation: `ranked_event_days <- event_days |> mutate(tmax_f = tmax * 9 / 5 + 32, weather_score = -abs(tmax_f - 72) - prcp) |> group_by(month, day) |> summarise(avg_score = mean(weather_score, na.rm = TRUE), avg_tmax_f = mean(tmax_f, na.rm = TRUE), avg_prcp = mean(prcp, na.rm = TRUE), years_observed = sum(!is.na(weather_score)), .groups = "drop") |> filter(years_observed > 0) |> arrange(desc(avg_score)) |> mutate(rank = row_number()) ranked_event_days`
2. `event_days |> head()` # Recommendation: dates with mild temperatures, low rain, and low variability
3. `event_days |> head()` # Recommendation: j
4. `event_days |> head()` # Recommendation: use sunnier days

Student responses

1. `event_days |> head()` `rain_ranking <- event_days |> mutate(rainy = prcp >= 1, temp_f = ((tmin + tmax) / 2) * 9 / 5 + 32) |> group_by(month, day) |> summarize(n_years = sum(!is.na(prcp)), n_rainy = sum(rainy, na.rm = TRUE), mean_temp = mean(temp_f, na.rm = TRUE), .groups = "drop") |> mutate(distance = abs(mean_temp - 72)) |> mutate(general_comparison = n_rainy + distance) |> arrange(general_comparison, na.rm=true) head(rain_ranking)`
2. `event_days |> head()` `rain_ranking <- event_days |> mutate(rainy = prcp >= 1) |> group_by(month, day) |> summarize( n_years = sum(!is.na(prcp)), n_rainy = sum(rainy, na.rm = TRUE), .groups = "drop" ) |> arrange(n_rainy)` `temperature_ranking <- event_days |> mutate(temp_f = ((tmin + tmax) / 2) * 9 / 5 + 32) |> group_by(month, day) |> summarize( n_years = sum(!is.na(temp_f)), mean_temp = mean(temp_f, na.rm = TRUE), .groups = "drop" ) |> mutate(distance = abs(mean_temp - 72)) |> arrange(distance) # Recommendation:`

Student responses

1. `event_days |> head()` `rule <- event_days |> mutate(score = if_else ( tmin > 15.5, tmax < 27, 1 )) rule |> head()` # Recommendation:
2. `event_days |> head()` `temperature_ranking <- event_days |> mutate(temp_f = ((tmin + tmax) / 2) * 9 / 5 + 32) |> group_by(month, day) |> summarize( n_years = sum(!is.na(temp_f)), mean_temp = mean(temp_f, na.rm = TRUE), .groups = "drop" ) |> mutate(distance = abs(mean_temp - 72)) |> arrange(distance)`
`rain_ranking <- event_days |> mutate(precip = ((tmin + tmax) / 2) * 9 / 5 + 32) |> group_by(month, day) |> summarize(n_years = sum(!is.na(temp_f)), mean_precip = mean(temp_f, na.rm = TRUE), .groups = "drop") |> mutate(distance = abs(mean_temp - 72)) |> arrange(distance) weather_score = rain_ranking + temperature_ranking` # Recommendation:
3. `event_days |> head()` `weather_score -> (((tmin + tmax) / 2) / precip) < 1000` # Recommendation:
4. `event_days |> head()` `mutate(score = as.integer( (tmin > 22) & (tmax < 30) ))` #

Student responses

1. `event_days |> head() mutate(score = if_else( (tmin > 20) & (tmax < 30), score = 1, score = 0 ) ) # Recommendation:`
2. `event_days |> head() day_ranking <- event_days |> mutate(temp_f = ((tmin + tmax) / 2) * 9 / 5 + 32) |> group_by(month, day) |> summarize(n_years = sum(!is.na(temp_f)), mean_temp = mean(temp_f, na.rm = TRUE), .groups = "drop") |> mutate(score = abs(mean_temp - 70) - precip) |> arrange(score) # Recommendation:`
3. `event_days |> head() event_days |> mutate(score = if_else(tmin >= 22 && tmax <= 30, 1, 0)) |> group_by(month, day) |> summarize(m = mean(score, na.rm = T)) # Recommendation:`
4. `event_days |> head() mutate_(score = as.integer( (tmin > 12) & (tmax < 30))) |> group_by_(month, day) |> summarize(mean(score))`

Student responses

1. `event_days |> head() mutate(avg= (tmin + tmax)/2, good_rain = prcp < 1.5, score = avg < 80 & avg > 6) |> group_by(month,day) |> summarize( n_years = sum(!is.na(temp_f)), mean_temp = mean(temp_f), .groups = "drop" ) |> arrange(event_days) # Recommendation:`
2. `event_days |> head() mutate(score = as.integer( (tmin > 20) & (tmax < 30) )) |> filter(month == 7) |> group_by_(month, day) |> summarize(mean(score)) |> arrange(desc(m)) # Recommendation:`
3. `event_days |> head() mutate(score = as.integer( (tmin > 22) & (tmax < 30))) |> group_by(month,day) |> summarize(mean(score))`
4. `event_days |> head() temperature_ranking <- event_days |> mutate(temp_f = ((tmin + tmax) / 2) * 9 / 5 + 32) |> group_by(month, day) |> summarize(n_years = sum(!is.na(temp_f)), mean_temp = mean(temp_f, na.rm = TRUE), .groups = "drop") |> mutate(distance = abs(mean_temp - 72)) |> arrange(distance) # Recommendation: it should be 1996/05/01`

Student responses

1. `event_days |> head()` `temperature_ranking <- event_days |> mutate(temp_f = ((tmin + tmax) / 2) * 9 / 5 + 32) |> group_by(month, day) |> summarize( n_years = sum(!is.na(temp_f)), mean_temp = mean(temp_f, na.rm = TRUE), .groups = "drop" ) |> mutate(distance = abs(mean_temp - 72)) |> arrange(distance) #`
Recommendation: May 1
2. `event_days |> head()` `temperature_ranking <- event_days |> mutate(temp_f = ((tmin + tmax) / 2) * 9 / 5 + 32) |> group_by(month, day) |> summarize( n_years = sum(!is.na(temp_f)), mean_temp = mean(temp_f, na.rm = TRUE), .groups = "drop" ) |> mutate(distance = abs(mean_temp - 72)) |> arrange(distance) #`
Recommendation: recommend

Student responses

1. `event_days |> head()` `temperature_ranking <- event_days |> mutate(temp_f = ((tmin + tmax) / 2) * 9 / 5 + 32) |> group_by(month, day) |> summarize( n_years = sum(!is.na(temp_f)), mean_temp = mean(temp_f, na.rm = TRUE), .groups = "drop" ) |> mutate(distance = abs(mean_temp - 72)) |> arrange(distance)`
`rain_ranking <- event_days |> mutate(rainy = prcp >= 1) |> group_by(month, day) |> summarize( n_years = sum(!is.na(prcp)), n_rainy = sum(rainy, na.rm = TRUE), .groups = "drop" ) |> arrange(n_rainy)` `rain_ranking` `weather_score <- event_days |> mutate(temp_f = ((tmin + tmax) / 2) * 9 / 5 + 32) |> mutate(w_score = ifelse( (rainy=5) & (temp_f < 75)& (temp_f > 60), 1, 0) )` `group_by(month, day) |> summarize(mean(score) )` # Recommendation:
2. `event_days |> head(5) |> filter(prcp != 0) |> mutate(score = abs(((tmin + tmax) / 2) * 9 / 5 + 32)-65) |> group_by(month, day) |> summarize ( mean_score = mean(score, na.rm = TRUE), .groups = "drop" )` `arrange(mean_score)` # Recommendation: may first

Student responses

1. `event_days |> mutate( temp_f = ((tmin + tmax) / 2) * 9 / 5 + 32, good_day = prcp < 1 & temp_f >= 68 & temp_f <= 76 )|> group_by(month, day) |> summarize(n_good = sum(good_day, na.rm = TRUE), .groups = "drop") |> arrange(desc(n_good))` # Recommendation: # August 8th, since it has the most favorable record.
2. `event_days |> mutate(prcp = prcp == 0) filter(event_days, prcp == TRUE) event_days` # Recommendation:
3. `event_days |> mutate(score = as.integer( (tmin > 12) & (tmax < 30) )) |> group_by(month, day) |> summarize(m = mean(score)) |> arrange(-tmin)` # Recommendation: # Based on the new score, recommended dates are around early may # event_days
4. `event_days |> mutate(score = as.integer( (tmin > 22) & (max < 30))) |> group_by(month, day) |> summarize(mean(score))` # Recommendation:
5. `event_days|> mutate(score = as.integer( (tmin > 12) & (tmax < 30)) |> group_by(month, day) |> summarize(mean(score)) |> arrange(m)` # Recommendation: weigh each value and combine to determine best day.

Student responses

1. `event_days$score = temperature_rank$distance event_days event_days_2 <- event_days |> group_by(month, day) |> summarize(n_years = sum(!is.na(prcp)),
n_rainy = sum(rainy, na.rm = TRUE), .groups = "drop") event_days_2 # Recommendation:`
2. `good_days <- event_days |> mutate(rainy = prcp >= 1, temp_f = abs((((tmin + tmax) / 2) * 9 / 5 + 32) - 72) |> group_by(month,day) # Recommendation:`
3. `mean(T, T, F, T)`
4. `new_ranking <- event_days |> mutate(rainy = prcp >= 1) |> mutate(temp_f = ((tmin + tmax) / 2) * 9 / 5 + 32) |> group_by(month, day) |> summarize(n_years =
sum(!is.na(temp_f)), mean_temp = mean(temp_f, na.rm = TRUE), n_years = sum(!is.na(prcp)), n_rainy = sum(rainy, na.rm = TRUE), .groups = "drop") |>
filter(mean_temp <= 10, n_rainy <= 10) head(new_ranking) # Recommendation:`

Student responses

1. `ranked_dates <- event_days |> mutate(tmax_f = (tmax * 9 / 5) + 32, day_score = -(prcp * 2) - abs(tmax_f - 75)) |> group_by(month, day) |> summarise(avg_score = mean(day_score, na.rm = TRUE), avg_prcp = mean(prcp, na.rm = TRUE), avg_tmax_f = mean(tmax_f, na.rm = TRUE), .groups = "drop") |> arrange(desc(avg_score)) avg_score # Recommendation:`
2. `score <- event_days |> group_by(month, day) |> mutate(score = as.integer((tmin > 12) & (tmax < 30))) |> group_by(month, day) |> summarize(m = mean(score)) |> arrange(-m) # Recommendation:`
3. `temperature_ranking <- event_days |> mutate(temp_f = ((tmin + tmax) / 2) * 9 / 5 + 32) |> group_by(month, day) |> summarize( n_years = sum(!is.na(temp_f)), mean_temp = mean(temp_f, na.rm = TRUE), ) temperature_ranking |> mutate(score = abs(mean_temp - 72) + prcp) arrange(score) # Recommendation:`

Student responses

1. `temperature_ranking <- temperature_by_date |> arrange(distance) head(temperature_ranking) event_days |> head()`
`temperature_by_date <-`
`temperature_by_date |> mutate(distance = abs(mean_temp - 72)) temperature_by_date |> select(month, day, mean_temp, distance) |> head()`
`mutate(temp_f = ((tmin + tmax) / 2) * 9 / 5 + 32) |> group_by(month, day) |> summarize( n_years = sum(!is.na(temp_f)), mean_temp = mean(temp_f, na.rm = TRUE), .groups = "drop" ) |> mutate(distance = abs(mean_temp - 72)) |> arrange(distance) |> summarize(mean(distance))` # Recommendation: October 5th has low rain and a good temperature of 72.

Student responses

1. `weather_ranking <- event_days |> mutate(temp_f = ((tmin + tmax) / 2) * 9 / 5 + 32, rainy = prcp >= 1) |> group_by(month, day) |> summarize(n_years_temp = sum(!is.na(temp_f)), mean_temp = mean(temp_f, na.rm = TRUE), n_years_rain = sum(!is.na(prcp)), n_rainy = sum(rainy, na.rm = TRUE), .groups = "drop") |> mutate(distance = abs(mean_temp - 72), temp_rank = min_rank(distance), rain_rank = min_rank(n_rainy), total_rank = temp_rank + rain_rank) |> arrange(total_rank) weather_ranking # therefore, recommend August 16th`

Choose a date using your rule. Give a number from your results that supports your choice. How does it compare with the two simple rules, and what would you still want to know before booking the picnic?

0 anonymous responses

No responses were submitted.

Choose one part of your rule that another group might reasonably disagree with. Predict how changing it would affect the ranking, then rerun the analysis. Does your recommendation change? What would you now tell the club?

0 anonymous responses

No responses were submitted.

Both pipelines run. What question does each answer? Explain which days contribute to each mean, rather than just describing the lines of code.

0 anonymous responses

No responses were submitted.

Would you expect the same food to be most popular in both groups? What would you calculate to decide whether one menu would suit both? Explain what an overall ranking of food choices might leave out.

0 anonymous responses

No responses were submitted.

The observed counts differ from the counts we constructed using the product of the marginals. Does this establish that food choice and birth continent are not independent in the population? Explain your reasoning. What else would you need to know before drawing that conclusion?

74 anonymous responses

Need formal hypothesis test

33

Students say differences alone don't prove non-independence and that a formal statistical test (e.g., chi-square, p-value) is needed to assess significance.

Randomness and noise may explain differences

13

Students attribute discrepancies to randomness, noise, or small sample size rather than true dependence.

Concludes dependence without caveats

9

A student asserts that differing observed and expected counts indicate the variables are not independent, without mentioning uncertainty.

Need joint distribution or matching-not-by-chance evidence

7

Says we need to know the joint distribution or evidence that the pattern isn't due to chance before concluding dependence.

Response themes for question 11

Random sampling and representativeness concerns**4**

Students point out the sample may not be random or representative, so conclusions about the population can't be drawn.

Concern about confounding or other factors**3**

Points out possible confounding variables or other unmeasured factors that could explain the observed differences.

Suggests data/table errors or covariance issues**3**

Student suggests that discrepancies might arise from errors in the table or covariance-related issues rather than dependence per se.

Other responses**2**

Student responses

1. Does not establish they are not independent. Need to run a chi squared test and actually get a p value with a test. We need further statistical testing
2. I don't think it would necessarily mean that they are independent. We would need to run further testing such as a p-test
3. if we find that the choice are independent then there is not a correlation between the two, a test should be conduct to verify this
4. it can't say that those two variable are not independent, need formal test
5. It does not. Randomeness and noise may explain the differences. We will need a hypothesis test because drawing the conc
6. Just because the observed counts vary from expected count. can't say its independent because you need p-value to see how "far off" each count is and take into account random variability
7. No it does not. Our job is to figure out if the difference is statistically significant, or just variance / noise.
8. No it doesn't mean they are not independent. We need an actual hypothesis test

Student responses

1. No, because the test would need to be run more than once for you to rule out independence. One survey isn't enough data to be statistically significant.
2. No, because there's always a bit of randomness that can lead to variation as well as possibly another factor that confounds the relationship.
3. No, it could be due to random chance alone that the counts done match up completely. I would like to see a p test as well to understand the statistical significance of the results we got
4. No, it does not establish that, you would still need more to go off of and cannot conclude if independent or dependent yet, would need standard dev, chi square etc
5. No, it does not establish this. We would need to get a p-value to see if things are in fact dependent
6. No, the results of one survey with 98 respondents can not prove that food choice and birth continent are independent or dependent, you would need a formal test with more responses to make a conclusion

Student responses

1. No, the tables would not match. There is a expected amount of variation between expected amount and observed amount. You need a formal statistical test to determine if the differences between the observed and expected amounts are statistically significant.
2. No, there could just be some sampling variance so the events could be independent, but the tables don't match up exactly
3. No, we are just looking at a sample (and getting exactly on the mark from samples is rare). So we should run a significance test of some sort to find convincing evidence
4. No, we have to see if that difference is statistically significant through a chi squared test/p-val test
5. No, we need to conduct a test to generate a p-value to see if the differences are statistically significant.
6. No, we would need a controlled test to establish that. Theres specific formulas for this that are not coming to mind right now
7. no, we would need a formal hypothesis test
8. No, with only this sample we cannot be sure that the factors are independent. We need multiple repeated samples and to run a test that returns a low p value.

Student responses

1. No, you need to do additional calculations on the product of the counts
2. No. random sampling variation can cause differences and you should perform a chi-square or simulation test and check the survey's representativeness before concluding dependence.
3. The counts are not expected to match perfectly. However to decide if the relation is independent or dependent, we need to do actual statistical testing
4. The observed counts will be different, but it is not evidence because we did not construct a t-test or other statistical test. We need to evaluate the evidence against the null hypothesis before drawing conclusions.
5. The tables would not match exactly even if the two variables were independent, because some variance is expected. You would need to do a statistical test to show that the two variables are independent.
6. There would have to be a much higher trend or correlation between food choice and continent

Student responses

1. They would not match exactly as the expectation disregards noise (assumed $E[\epsilon] = 0$). The observed will not necessarily match the expectation. I suppose we'd need a formal hypothesis test to check the sample difference.
2. We have a much smaller sample of people from Asia compared to north America. Also, since this is random it should be exact it just needs to be close.
3. We still need to perform a Chi-square independence test to know if they are independent. A mere difference won't tell.
4. We would need a larger sample size/one more reflective of the general population in order to conclude that food choice and birth continent are not independent. If we keep the same sample, we would still need to run a chi-squared test to check for differences in observed/expected counts
5. You would need a formal hypothesis test in order to confirm independence, there is a lot of randomness and variability that could be accounting for this difference since the numbers are so small.

Student responses

1. Construct a residual plot to see if there are any confounding variables and another simulation to see if the sample is truly random and independent, chi square
2. I do not think that they necessarily have to match. We keep emphasizing that we are working with an extremely noisy set of data so that may contribute to it.
3. It does not have to exactly match the exact number to be independent. On the other hand, when it matches exactly, it does not mean it is not dependent.
4. no you wouldnt as we would assume some degree of randomness among the observed points.
5. No, because of randomness.
6. No, it doesn't have to be exact
7. No, since we are only using a small sample size, we should try another way of calculating
8. Not necessarily, it would depend on how much they vary from the expected, and its that's beyond reasonable variance.
9. The first table was counting in discrete way, where the second table is computed in continuous way. The noise may also change the results.

Student responses

1. The sample size of survey responses isn't sufficient enough to determine that they're independent
2. there could be other factors in play, maybe people who take data sci classes are more into sushi. or the sample size is not big enough
3. there is still some variability so you cant expect them to match exactly
4. this does not establish that food vhoive and birth continent are not independent

Student responses

1. food choice and birth continent are not independent and would need to confirm with h tets
2. The differing counts suggest that food choice and birth continent are not independent in the population, because the real counts do not match the scenario where they are independent, implying that birth continent affects food choice.
3. This establishes that food choice is *likely* not independent of birth continent. We would have to run a separate t-test in order to see if there is a relationship between the two groups.
4. Ya it means it snot inedpedent. THis is because where you born can affect ur food prefernces as food you ate growing up you willbecome more accustomed to and enjoy eating more. We would need ot know unfenerecn etst
5. Yes
6. yes it does because it is standardized
7. Yes it does establish that they are not independent. This is because it is clear that there is a difference between them when looking at the product of marginals. However, we would need to run a t-test of some sorts to verify.

Student responses

1. Yes this establishes that the food choice and the birth content are not independent in the population. We would need a formal hypothesis test to help draw this conclusion
2. Yes, because the popular food choice is the same in the observed counts, if they were dependent they would be close to expected counts

Student responses

1. Before drawing that conclusion, you would need to know that the variables are matching up NOT by chance
2. It does not establish that food choice and birth continent are not independent in the population. The expected counts are just probabilities but due to external factors the expected counts won't always match. That doesn't mean that the variable are independent. In order to know independence you would need to run tests that determine associations (p-value).
3. No I dont think it establishes that food choice and birth contienetn are not indepeendent, Im not entirley sure on what else we will need to know, but I think we would have to re apply the rule?
4. no, they can be dependent on each other, but not strictly the case. Need bigger pool of info
5. No. Food choice and birth continent each have their own weights that contribute to the product of the marginals, as seen by the weights they carry. I'm not sure what else would be needed.
6. They establish that they are independent since food preferance could be random

Student responses

1. This does not establish that food choice and birth continent are not independent in the population because we do not know if the data satisfies the necessary assumptions, such as the normality assumption. We would need to know how the data was collected to know whether this difference establishes a lack of independence.

Student responses

1. I don't think that given our data, the variables are not independent because this class is technically not a random sample, so we cant generalize to the public with our class data.
2. No, as it is not a randomly selected dataset
3. No, this case it was the same, but it doesn't have to be.
4. No, we are doing random sampling, so the observed would have different to what we calculated

Student responses

1. It leads us towards that statement, but we don't know if there are any potential confounding variables that could correlate them.
2. No. There might be other factors we did not compute in this data set that might be affecting between two choices
3. This does not establish that food choice and birth continent are not independent in the population. You need to run the test to understand the margins the cause it to be independent rather than dependent.

Student responses

1. im not sure
2. No, since the popular food choice is the same for both groups but it is not always the case.
3. yes, with covariance, it could be some error in the table

Sushi is the most popular choice in both groups, but its share is 55.6% in one and 32.5% in the other. The test of the full table has a p-value around 0.18. What would you recommend ordering? Use the observed choices and explain what the test does, and does not, establish.

0 anonymous responses

No responses were submitted.

**What do these percentages actually describe?
Which votes were removed too soon, and how
would you repair the calculation?**

0 anonymous responses

No responses were submitted.

Suppose an outdoor event must take place on a day in September through November, and can tolerate cooler weather. Describe how you would change our weather analysis. Which steps would stay the same, and why would adding November require going back to the daily data?

0 anonymous responses

No responses were submitted.