How do we recommend movies?

Relational data · Week 3, Lecture 5

https://live.ds306.org

September 15, 2026

What should we watch next?

Your roommate loved *Barbie*. You're thinking of putting on *Oppenheimer*.

The data

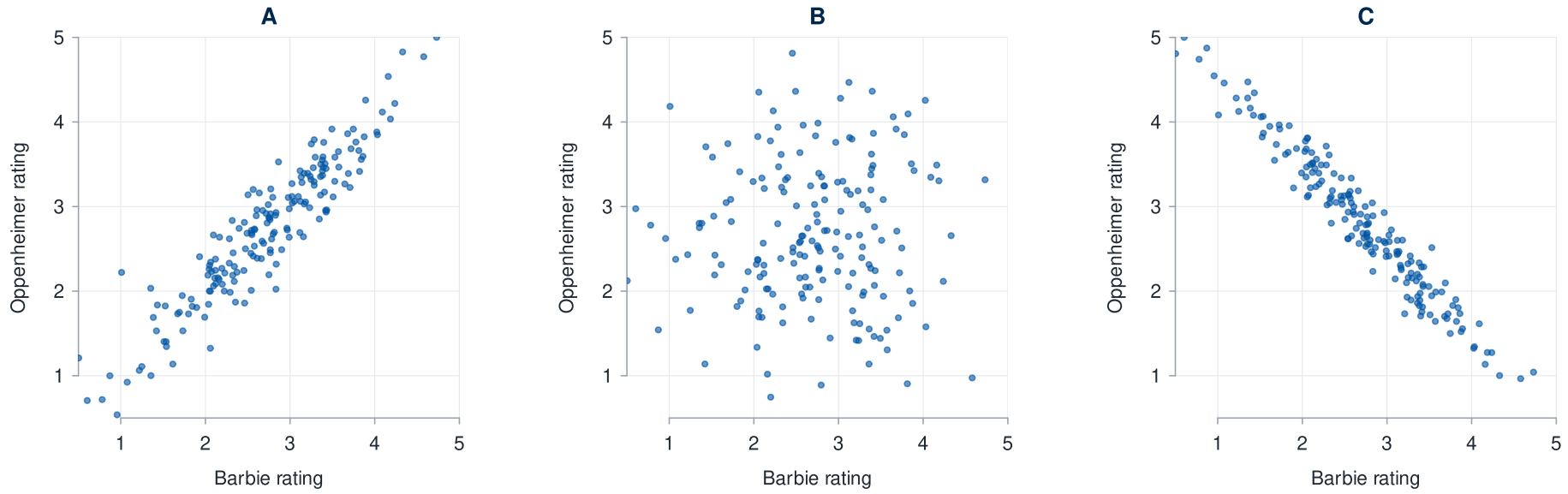
- MovieLens (data documentation).
- Offline, we extracted a subset of the full MovieLens data for our analysis.
- The full release has 200,948 users and about 32 million ratings.
- For Q1, our extract contains 939 people who rated *Barbie*, and their ratings of 1,904 films.

How will we use the data?

- We'll compare the ratings of people who rated **both films**.
- Let's start with a scatter plot: each person's *Barbie* rating on the horizontal axis and their *Oppenheimer* rating on the vertical axis.
- We want to see how accurately we might predict someone's *Oppenheimer* rating from their *Barbie* rating.

Where would predictions be more accurate?

Here are three imaginary sets of ratings. Each point represents one person.



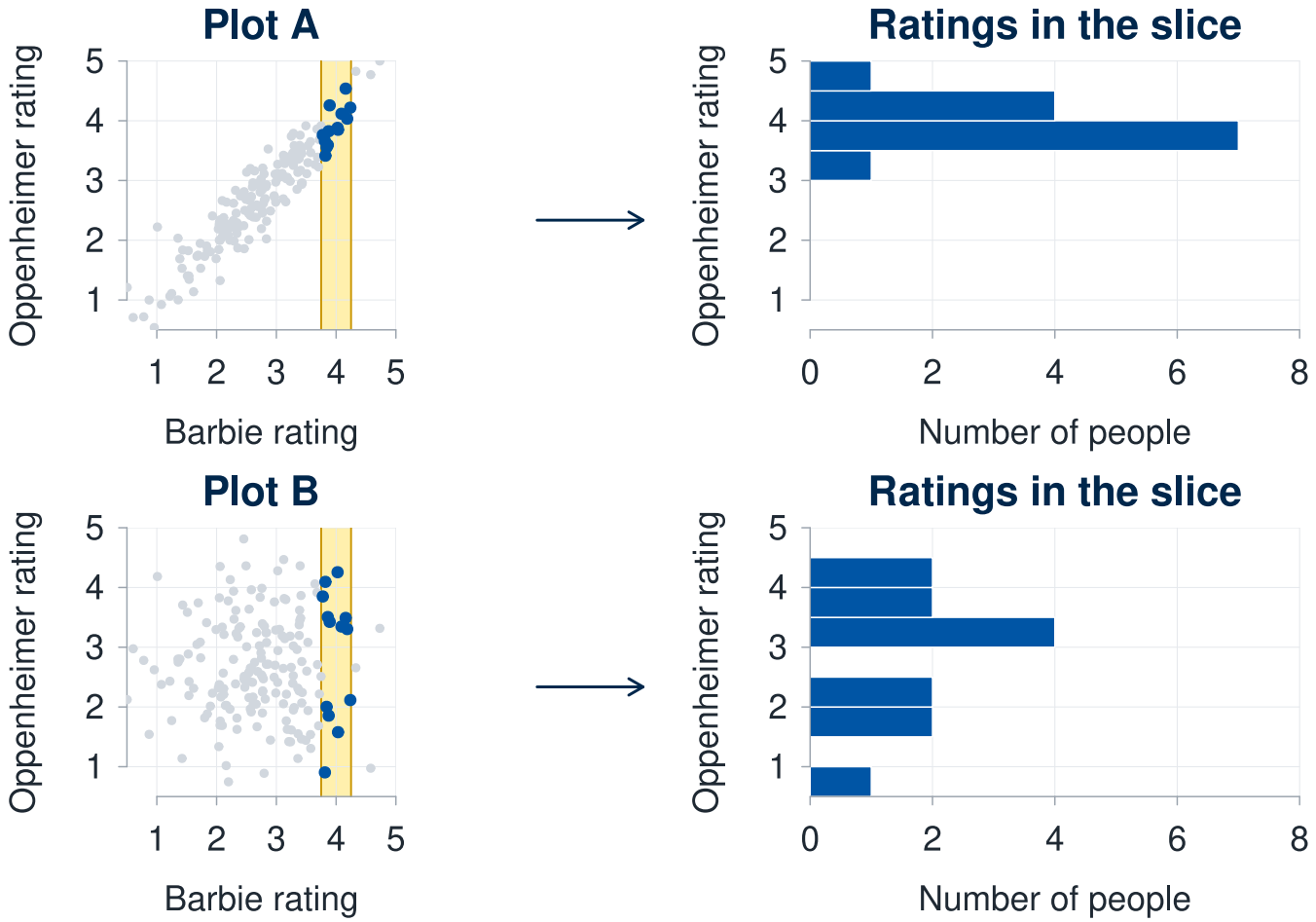
> Someone gave Barbie four stars. Which plots let you predict their Oppenheimer rating more accurately? Why?

Signal and noise

- The trends in A and C are **signal**: knowing the *Barbie* rating helps us predict the *Oppenheimer* rating in either case.
- The variation left among people with the same *Barbie* rating is **noise**, relative to the information we're using.
- B has much more remaining variation. Knowing someone's *Barbie* rating leaves substantial uncertainty about their *Oppenheimer* rating.

Prediction error

Slice the plots to include only people who rated *Barbie* around four stars:



How accurate could we get?

- Suppose we knew the exact average *Oppenheimer* rating for people who give *Barbie* four stars.
- Those people would still give different *Oppenheimer* ratings. We could predict the group average and still be wrong about an individual.
- Using *Barbie* ratings alone, that remaining noise sets a lower limit on our **average prediction error** for new people.
- More observations help us estimate the average, but will not reduce individual variability.
- Another useful predictor, such as ratings of similar films, could explain some of the remaining variation.

The ratings

```
ratings |> head(5)
```

- Each row is one person's rating of one movie.
- `user_id` identifies the person; `movie_id` identifies the movie.
- `rating` is the number of stars they gave it. The movie titles are stored separately.

The movies

```
movies |>
  select(movie_id, title, year) |>
  head(5)
```

- Each row describes one movie.
- `movie_id` lets us look up its title and release year.
- These are the first five rows. *Barbie* and *Oppenheimer* appear farther down the table.

What table would let us draw the plot?

- Each point will represent one person: their *Barbie* rating on the horizontal axis and their *Oppenheimer* rating on the vertical axis.
- We need a table like this. These are three people's actual ratings:

```
both |>
  filter(user_id %in% c(28, 874, 1963)) |>
  arrange(user_id)
```

- Each row contains **one person's two ratings**. Let's work out how to build it.

How will we carry out the comparison?

```
ratings
movies
```

> Starting with `ratings` and `movies`, what would you need to do to be able to make the scatter? (Explain in words if you don't know the code.)

Our plan

1. Attach movie titles to the ratings.
2. Make a table for each film, then join them to put each person's ratings side by side.
3. Plot the paired ratings to see how much *Barbie* ratings tell us about *Oppenheimer* ratings.
4. Calculate the fraction of *Barbie* fans who liked *Oppenheimer*, then apply our decision rule.

- For this example, four stars counts as liking a film.
- We'll recommend it if at least 75% of the *Barbie* fans who rated both films liked *Oppenheimer*.

Relational data

- So far, we've mostly worked with one table at a time. Our movie question needs information from two tables.
- `movies` describes the films. `ratings` records what people thought of them.
- These are **relational data**: related tables whose rows can be connected using identifying columns called **keys**.
- A **relational database** stores and manages these tables.

Why keep movie information separate?

Why doesn't MovieLens store all the movie information directly in the `ratings` table like this?

```
# A tibble: 8 × 6
  user_id rating movie_id title          year genres
    <dbl>  <dbl>    <dbl> <chr>         <dbl> <chr>
1      28    5     288513 Barbie (2023)  2023 Comedy
2      79    4.5   288513 Barbie (2023)  2023 Comedy
3     326    4     288513 Barbie (2023)  2023 Comedy
4     458    4     288513 Barbie (2023)  2023 Comedy
5     872    3     288513 Barbie (2023)  2023 Comedy
6     874    4.5   288513 Barbie (2023)  2023 Comedy
7    1963    3     288513 Barbie (2023)  2023 Comedy
8    2343    5     288513 Barbie (2023)  2023 Comedy
```

Why keep movie information separate?

- Hundreds of people rated *Barbie*. Putting everything in one table would repeat its title, release year, and genres in every rating row.
- If we correct a movie's information, we can change one row in `movies`.
- We can add a movie to `movies` before anyone has rated it.
- A new rating only needs the person's ID, the movie's ID, and the number of stars.

For our analysis, we'll bring the title beside each rating using a **join**.

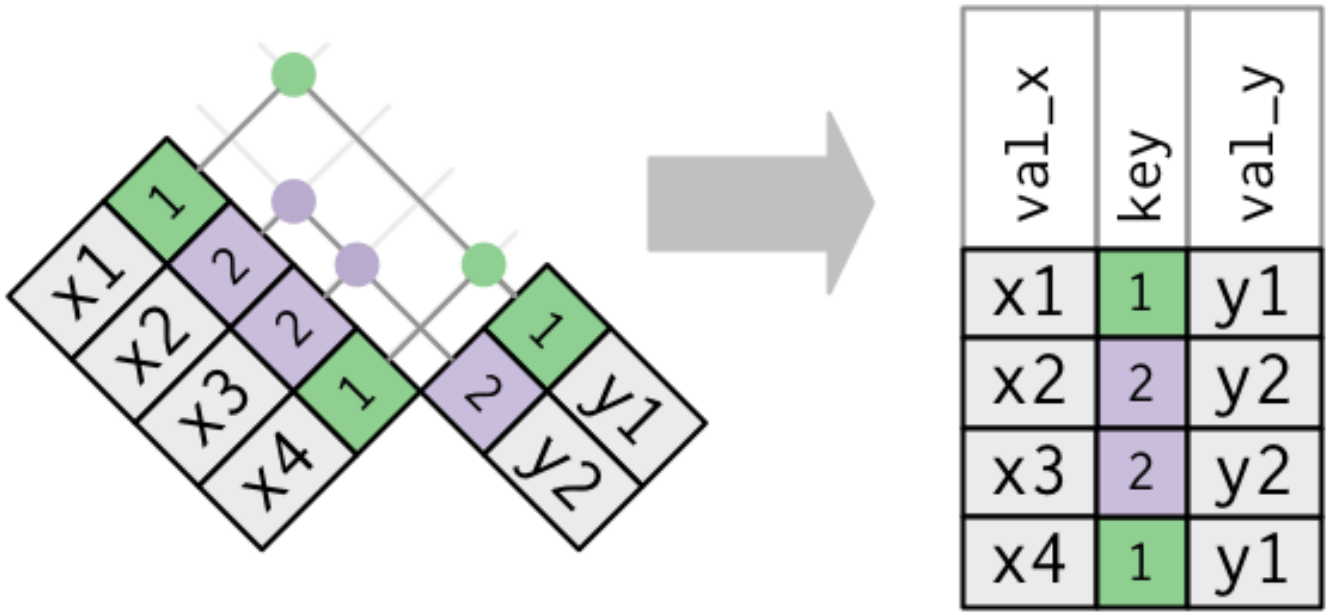
One movie can have many ratings

- In `movies`, each `movie_id` identifies one row. It is the **primary key**.
- In `ratings`, `movie_id` refers to that movie's row in `movies`. Here it is a **foreign key**, and it can repeat.
- This is a **one-to-many relationship**: one movie can match many rating rows.

For example, *Barbie*'s ID appears once in `movies` and once for each person who rated it in `ratings`.

How a key connects the rows

Think of x as ratings and y as movie information.



val_x	key	val_y
x1	1	y1
x2	2	y2
x3	2	y2
x4	1	y1

Each rating receives the information for its movie. Multiple ratings can look up the same movie.

Source: R for Data Science, Wickham and Grolemund.

Step 1: attach movie titles

```
ratings |> head(5)
```

- Each row is one person's rating of one movie.
- `user_id` identifies the person; `movie_id` identifies the movie.
- To find the *Barbie* ratings, we need to know which movie each ID refers to.

Titles live in another table

```
movies |>
  select(movie_id, title, year) |>
  head(5)

movies |> filter(title == "Barbie (2023)")
```

- Each row describes one movie.
- The same `movie_id` refers to the same movie in both tables.

Match a few rows by hand

Here are two rows from `ratings`:

```
mini_ratings
```

> Which movies did this person rate, and how many stars did they give each? Use `filter()` on `movies` to look up each movie ID. Explain which column connects the two tables.

Let R do the matching

```
mini_ratings |>
  left_join(movies, by = "movie_id") |>
  select(user_id, title, rating)
```

- Start with `mini_ratings`.
- Match its `movie_id` values to `movies`, and attach the movie information.
- Keep the person's ID, the title, and the rating. We still have two rows.

Attach titles to all the ratings

```
named_ratings <- ratings |>
  left_join(movies, by = "movie_id")

named_ratings |>
  select(user_id, title, rating) |>
  head(5)
```

Each row is still one person's rating of one movie.

Did the join behave as expected?

```
nrow(ratings)
nrow(named_ratings)

named_ratings |> filter(is.na(title)) |> head(3)
```

- Every rating should match exactly one movie, so the row count should stay the same.
- Every movie ID in this extract has a matching title.

Step 2: put the ratings side by side

Our next **subproblem** is to put each person's two ratings in one row.

After attaching titles, user 874's ratings occupy two rows:

```
named_ratings |>
  filter(user_id == 874, movie_id %in% c(287699, 288513)) |>
  arrange(movie_id) |>
  select(user_id, title, rating)
```

For the scatter plot, we need one row with both coordinates:

```
both |> filter(user_id == 874)
```

How would you solve this subproblem?

We have `named_ratings`, with one row for each person's rating of a film. We want one row per person, with their two ratings side by side.

> How would you build that table using the tools we've learned? Describe your steps in words. How will you make sure the two ratings in each row belong to the same person?

Make a table for each film

1. Keep the *Barbie* ratings. Keep `user_id` and name the rating column `barbie_rating`.
2. Do the same for *Oppenheimer*, naming its rating column `oppenheimer_rating`.
3. Join these two tables using `user_id`. Each person's two ratings will appear in the same row.

Each table will have one rating per person. The shared `user_id` will let us match them.

First, keep the Barbie ratings

```
named_ratings |>
  filter(title == "Barbie (2023)") |>
  head(4)
```

- Each remaining row is a different person's rating of *Barbie*.
- We need to keep the person ID so we can find that person's *Oppenheimer* rating.

Save a table of Barbie ratings

```
barbie <- named_ratings |>
  filter(title == "Barbie (2023)") |>
  select(user_id, barbie_rating = rating)

barbie |> head(4)
```

- Inside `select()`, `barbie_rating = rating` keeps the `rating` column and names it `barbie_rating`.
- Every row now contains a person's ID and their *Barbie* rating. We save this table as `barbie`.

Your turn: make the Oppenheimer table

```
# Change this example to find Oppenheimer's ratings.
named_ratings |>
  filter(title == "Barbie (2023)") |>
  select(user_id, barbie_rating = rating) |>
  head(4)
```

> Change the movie title and the rating column's new name. Submit your two changed lines. Why should we keep user_id?

Save the second table

```
oppenheimer <- named_ratings |>
  filter(title == "Oppenheimer (2023)") |>
  select(user_id, oppenheimer_rating = rating)

oppenheimer |> head(4)
```

We now have one table of *Barbie* ratings and another of *Oppenheimer* ratings.

How should these tables match?

Here are some rows from each table:

`barbie`

```
barbie |>
  filter(user_id %in% c(28, 874, 1963)) |>
  arrange(user_id)
```

`oppenheimer`

```
oppenheimer |>
  filter(user_id %in% c(28, 874, 1963)) |>
  arrange(match(user_id, c(874, 1963, 28)))
```

> Which Oppenheimer rating belongs beside each Barbie row, in order? Explain how R should find the matches when the rows are in different orders.

Match the same person across movies

```
paired <- barbie |>
  left_join(oppenheimer, by = "user_id")

paired |> head(6)
```

- Start with everyone in `barbie`, and match `user_id` values in `oppenheimer`.
- Each row now describes one person, with a separate rating column for each film.
- User 874's ratings, 4.5 and 5, are now side by side. They give us one point on the scatter plot.

Where did these NAs come from?

```
paired |>
  filter(is.na(oppenheimer_rating)) |>
  head(3)
```

> What do we know about these people's ratings, and what don't we know? Would replacing the missing values with zero be reasonable? Explain.

Compare people who rated both

```
both <- paired |>
  filter(!is.na(oppenheimer_rating))

nrow(paired)
nrow(both)
```

- There are 939 *Barbie* raters, of whom 471 also rated *Oppenheimer*.
- Only those 471 people contribute to the comparison.

Step 3: plot the paired ratings

```
ggplot(both, aes(x = barbie_rating, y = oppenheimer_rating)) +
  geom_point(alpha = 0.3, size = 3)
```

There are 471 people in `both`. Where are all their points?

Count each pair of ratings

```
rating_counts <- both |>
  count(barbie_rating, oppenheimer_rating)

rating_counts |> top_n(5)
```

- Group together people who gave the same pair of ratings, then count them.
- Each row now represents a pair of ratings. For (4, 4), n is 45.

A 2D heatmap

```
ggplot(rating_counts, aes(x = barbie_rating, y = oppenheimer_rating)) +
  geom_tile(aes(fill = n), color = "white") +

  scale_fill_gradient(low = "white", high = "#0055a5", limits = c(0, NA)) +


  theme_minimal(base_size = 18)
```

Darker cells represent more people. Blank cells have a count of zero.

Interpreting the heatmap

> Does this plot indicate that people who like *Barbie* also tend to like *Oppenheimer*? Dislike? Explain your reasoning.

Step 4: calculate the fraction

Among people who liked *Barbie*, what percentage also liked *Oppenheimer*? Use `both`, which contains people who rated both movies. A rating of at least four stars counts as liking a movie.

```
recommendation <- both |>
  filter(___) |>
  summarize(people = n(), liked = sum(___)) |>
  mutate(percent_liked = ___)

recommendation
```

> Replace each ____ and run your code. Submit your completed code and the percentage you found.

Back to your roommate

Our rule was to recommend *Oppenheimer* if at least 75% of the *Barbie* fans who rated both movies gave it four stars or more.

> Would you recommend *Oppenheimer* to your roommate? Use the fraction and the heatmap to explain your answer. What uncertainty remains?

The whole analysis

Click a step to reveal the code we used.

```
library(tidyverse)
movies <- read_csv("movies.csv", show_col_types = FALSE)
ratings <- read_csv("ratings.csv", show_col_types = FALSE)


named_ratings <- ratings |>
  left_join(movies, by = "movie_id")


barbie <- named_ratings |>
  filter(title == "Barbie (2023)") |>
  select(user_id, barbie_rating = rating)

oppenheimer <- named_ratings |>
  filter(title == "Oppenheimer (2023)") |>
  select(user_id, oppenheimer_rating = rating)


paired <- barbie |>
```

How to approach a complex problem

- Decide what answer would help you make the decision. Here, we chose a percentage and a plot.
- Work backward: what table would give you that answer? How could you build it from the data you have?
- Write the steps in words. Break a difficult step into smaller ones, then translate them into code.
- You can give AI that plan and ask for help with one step at a time. Run the code, inspect its output, and revise the plan when needed.
- Be able to explain each step and how you checked it. Here, that includes matching the same person and choosing the right denominator.

Q2: Is the sequel actually better?

Choose a pair (work in groups!):

- *Shrek* (2001) and *Shrek 2* (2004).
- *Top Gun* (1986) and *Top Gun: Maverick* (2022).
- *Spider-Man: Into the Spider-Verse* (2018) and *Spider-Man: Across the Spider-Verse* (2023).
- Which film did people rate more highly?
- In terms of statistical evidence, what would it take to convince you that the sequel is or is not better than the original?

Data for the sequel comparison

- This extract contains every rating of these six films in MovieLens 32M.
- `sequel_data` contains the ratings with movie information attached, just like `named_ratings` in Q1.

```
sequel_data |>
  select(user_id, title, rating) |>
  head(6)
```

Work backward from your answer

Compare people who rated both films. For the statistical comparison, use the average difference in stars: sequel minus original.

> What table would let you estimate the average rating difference between two different films and run a `.test()`? Write a short plan for how you would construct it starting from `sequel_data` by working backwards.

A table we could use

These are three people who rated both Shrek films:

```
sequel_example
```

- Each row contains one person's two ratings.
- To build a row, we need to find that person's rating in each film's table.
- Those two tables can come from filtering `sequel_data`.

Write the steps before the code

1. Collect the original film's ratings and the reviewers' IDs.
2. Do the same for the sequel.
3. Match the reviewers and keep those with both ratings.
4. Calculate sequel minus original for each person.
5. Estimate the mean difference and assess its uncertainty.
6. Explain which film was rated higher and how convincing the evidence is.

Check that these steps would produce the table and answer you described.

Build the analysis

We'll write one step at a time, check its result, then close the fold before moving on. What should the next step produce?

```
original_title <- "Shrek (2001)"
sequel_title <- "Shrek 2 (2004)"


# Write this step below.


# Write this step below.


# Write this step below.


# Write this step below.

# Print the current result here.
```

Check your results, then report them

Submit the film pair your code, and the resulting table. Which film had the higher mean rating? Can you use `t.test()` to construct a a confidence interval for the difference? What's the p-value?

Week 3, Lecture 5

Closed September 15, 2026 at 3:47 PM

12 class questions · 755 anonymous responses · 0 student questions

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

Someone gave Barbie four stars. Which plots let you predict their Oppenheimer rating more accurately? Why?

85 anonymous responses

Prefer plots with clear trend/low variance — 43

Students say plots A and/or C are best because they show clear correlation and less variability, making Oppenheimer rating easier to predict from Barbie's four stars.

Favor plot C for strongest correlation — 24

Student identifies plot C specifically as the most useful because it appears narrow/skinnier and shows the strongest association among points.

Prefer B or no correlation reason — 11

Students argue plot B (or lack of correlation) is preferable or that ratings need not be related because movies differ in genre; they therefore don't expect predictive power from A/C.

Response themes for question 1

Prefer plot A due to axis choice — 4

Student prefers plot A specifically because Barbie is used as the independent variable (appropriate x-axis), implying it aids prediction.

Other responses — 3

Student responses

1. A & C as there appears to be a grouping of ratings given Barbie 4 stars. Whereas for B, it seems to be arbitrary for a given score.
2. A and C allow you to predict Oppenheimer rating more clearly because the data shows a clear association and is more tightly packed. Barbie rating corresponds with Oppenheimer rating for these two.
3. A and C are both helpful because the data is linear enough to make predictions
4. A and C because there seems to be a linear relationship between the two variables allowing for easy predictions
5. a and c cuz they have a clear pattern to them lolz
6. A and C let you predict someone's rating for Oppenheimer based off of their rating for Barbie most accurately because these plots have low variability compared to B.
7. A and C since they have a more clear association
8. A and C would be the most accurate becasue there are clear correlations and patterns while plot B is very scattered
9. A and C would help me to get an accurate oppenheimer ranking because they show strong coorelation between the ratings and B seems like a random scatter.

Student responses

1. A and C, because they show a clear correlation.
2. A or C because there is a clear relationship between the ratings in the scatterplots, while B is just a bunch of random points, which wouldn't help with predicting the Oppenheimer rating.
3. A or C because they all show strong correlation and would give a prediction with more confidence
4. A or C, they have a clear linear relationship between the 2 variables
5. B because theyre not related
6. c because there is the least variation in the data
7. Either A or C, because they show a linear relationship between the two movies' ratings
8. Either plot A or C because there is a clear correlation indicating the Oppenheimer choice. The randomness within B's plot makes it impossible to observe any relationship.
9. either plot a or c because those show the most clear correlation between a given barbie and oppenheimer vs plot b, which is just scattered with no real correlation.

Student responses

1. I think either plot A or C since the data seems to show a more linear relationship between the two movie ratings instead of the randomness in plot B
2. I would find A and C most helpful, because there is a clear trend between the barbie and oppenhimer rating.
3. I would find A and C the most valuable because there is a clear trend and not as much variance compared to B
4. Only A and C would be reasonable scatterplots that would allow you to draw conclusions about a relationship between Oppenheimer and Babie rating scores. The B graph provides no information to make an accurate prediction.
5. Plot A and C can. Because they showed a trend of that.
6. Plot A and C. Because they should clear tredn/relationship for the two variables, while plot B's variation for a fixed variable on the other varaible is much larger.
7. plot C because the variation is less then plot A and there is a correlation between barbie and Oppenheimer ratings

Student responses

1. Plot C would allow for an accurate rating of Oppenheimer, along with plot A. This is because there is a strong relationship between the two variables, so it is easier to predict the rating of Oppenheimer compared to plot B.
2. Plots A & C would accurately tell us an accurate prediction because there is a clear linear correlation between the two variables. Plot B is a very sparse, almost random looking graph, meaning there is no accurate way to predict one from the other.
3. Plots A and C are useful for predicting oppenheimer ratings from barbie ratings because they show a trend in the data while plot B shows no trend and the points are just randomly scattered.
4. plots A and C because they show more of a clear relationship between how viewers perceive both movies
5. Plots A and C let us predicts oppenheimer rating more accurately. This is because for those two graphs there is a clear correlation between barbie and oppenheimer ratings.
6. Plots A and C show a higher correlation between the two and would make it easier to predict the Oppenheimer rating more easily

Student responses

1. Plots A and C will help figure out the rating more accurately because they demonstrate a stong correlation
2. Plots A and C would allow us to estimate an Oppenheimer rating as they have a clear relationship between the variables.
3. Plots A and C would be helpful in predicting Oppenheimer ratings. There is a clear trend with these plots. In plot A, Oppenheimer rating appears to increase with Barbie rating. The opposite is true for plot C. Plot B is too scattered to see any real trend, and so it doesn't help us predict ratings.
4. Plots A and C would be helpful in predicting the Oppenheimer rater more accurately because those plots have a clear trend to show the prediction whereas plot B is scattered and would not be a good predictor
5. Plots A and C would be useful for predicting a person's Oppenheimer rating, because they show that the data has a fairly strong linear relationship.
6. Plots A and C would let us predict the Oppenheimer rating more accurately because the points follow a clear pattern. Plot B is more scattered, so the Barbie rating does not tell us as much about the Oppenheimer rating.

Student responses

1. Plots A and C wpuld best identify relations between ratings of Openheimer and Barbie. Plot B is very scattered and doesnt have a general trend, while plots A/C are not very scattered and follow a "line of best fit". PLot A is an upward trending relationship while plot C is a downward trending relationship between the likeness of the two films
2. Plots A and C, this is because to predict the Oppenheimer rating, it would be helpful if we can observe a strong correlation between the two movie ratings. Plot A displays a strong positive correlation, and Plot C displays a strong negative correlation, whereas Plot B doesn't really provide a correlation whatsoever.
3. Plots A and C. Both show a linear relationship between the two variables which allows us to make predictions due to covariance.
4. The first and third plots allow you to predict the oppenheimer rating pretty precisely based on the barbie rating, since there is a clear positive and negative corellation with those two. The middle one there appears to be no corellation, so prediction would be difficult

Student responses

1. The first and third, a and c, are the most useful as the data shows a clear trend I can follow and use to predict the rating they gave oppenheimer.
2. They all would but A/C is more easier to tell what the predicted Oppenheimer rating would be since the scatterplot has more of a linear trend to it.

Student responses

1. A and C, the dots follow a tighter line on the scatterplot, indicating a stronger relationship and more accurate predictions
2. Both A and C would allow me to predict how the Barbie raters rate Oppenheimer. as both of them have a general linear trend. C is much more accurate than A, but both still have that linear trend that is noticable.
3. C because the correlation is higher between the two
4. c gives a precise pattern
5. c or a both seem useful but c seems more realistic, creates some sort of equation so you can find the barbie rating on x axis and trace it up to the line and trace that to y axis to see what opp rating would be
6. C will be better, cause the dots is form a line
7. C, least variance. Very linear
8. C, since it is skinnier and shows a higher correlation among the points
9. Choice C is the best for predicting data because it is extremely linear with a very little variation.

Student responses

1. I think C gives the tightest grouping for linear regression/least error from a derived association, thus, assuming a true random sample, this would yield the best prediction.
2. I think the first and third one would be that correlation makes more sense to me
3. I would find C to be the most helpful in determiningg what rating that person is likely to give to Oppenheimer. This is because there is a clear shape to the scatter plot and we can use that pattern to make an educated guess of what is most like.y
4. Plot c because does
5. Plot C because it has a stronger association between the 2 variables. I would expect the r to be greater in magnitude allowing us to predict better
6. Plot C because it shows how people ewho liked barbie didnt like oppenheimer and vice versa
7. Plot C because Plot B is too scattered and Plot A is not showing the right relationship
8. Plot C would be the most useful, since it demonstrates the strongest relationship between the two ratings (scatterplot most closely resembles a line)

Student responses

1. Plot C would give a more accurate Oppenheimer rating because the correlation value r for this plot is the highest. So, the two variables have the highest correlation in this plot.
2. Plot C would provide the most accurate rating since the scatterplot has such a strong negative correlation with little spread.
3. Plot C. Plot B doesn't have a clear trend while both plot A and plot C do. Plus, plot C has more datapoints at Barbie rating=4.
4. plots C i guess since its just better
5. the third plot because it has the least amount of variation making it easier to predict Oppenheimer's rating
6. The third plot can most accurately predict the score, since the data points show the most clear linear relationship between the two ratings.
7. While A and C are similar, I would say C lets me predict their oppenheimer rating most accurately because the points are more condensed. graph B is more like a scatter plot and is not useful for predicting one rating given another.

Student responses

1. B because people have different views on barbie and Oppenheimer
2. B plot seems more accurate, because it may not have too strong correlation between these two movies
3. I am honestly not too sure what the answer is here. The A and C distributions are kinda like mirrors and show a trend line but this depends on whther there is actualy a trend line, but the B plot is a random distribution so it honestly doesnt fully reflect anything and is random so maybe b
4. I believe the plot B might show more accurate relationship between these two movies. A and C both are really linear relationship, feeling they are more extreme.
5. I guess B because thpugh there are stereotypes that people who like Barbie would not like movies too strict and boring, but good movies are good movies and people should try to be fair when rating even they don't like it
6. I guess c, since Oppenheimer and Barbie are two totally different types of movies, their ratings should be negatively related.

Student responses

1. I predict that plot B predicts the Oppenheimer rating more accurately, because if a large enough variety of people answer the survey, there should be no true correlation between ratings of the two films. Just because someone likes Barbie, it does not necessarily mean they will dislike Oppenheimer.
2. I think that graph B will most accurately represent this, because the movies are not related to each other, so people will have different ratings of the movie separately
3. I think the correct answer is B because the genre is completely different yet they both got awards. So we cant assume if one people like barbie the person will like oppenheimer or not like it
4. Plots A and C allow me to predict their Oppenheimer rating more accurately because there is a clear trend in both of these plots, unlike in Plot B.
5. third one because someone likes barbie may not like the other one

Student responses

1. A because it's linearly positive
2. plot a because it uses barbie as the independent variable
3. Plot A, babrbie should be x varibel and openherimer rayer should be the y variable the one affecting it also shows linear corerlation so easier ot predict oppenherimer rating thatn plot B
4. plot A, since we know that they gave barbie 4 stars, you need to look at oppenheimer rating as a response to that information as the y variable.

Student responses

1. 1 or 3 are equally easy to predict
2. A or C since there is positive or negative relation between Barbie and Oppenheimer.
3. I think those are all the same

Starting with `ratings` and `movies`, what would you need to do to be able to make the scatter? (Explain in words if you don't know the code.)

88 anonymous responses

Match ratings per user for two movies — 27

Identify ratings for both target movies for the same users so you can plot one user's rating of one movie against their rating of the other.

Filter ratings for specific movies — 25

Select ratings rows that correspond to the movie IDs for the movies of interest (e.g., Barbie and Oppenheimer) before plotting.

Join ratings with movies by movie ID — 11

Describes merging the ratings and movies tables on the movie ID to associate ratings with movie titles (so you can plot ratings by movie).

Plot using ggplot or similar — 7

Mentions using a plotting library/tool (ggplot) to create the scatterplot after preparing the data.

Response themes for question 2

Unclear or non-informative — 7

Responses that are too brief or do not state a concrete step toward creating the scatter (ambiguous single-character or unclear answers).

Remove missing values — 5

Suggests filtering out NA/missing entries before plotting so the scatter uses complete data points.

Create paired values per person for plotting — 3

Construct a combined value or paired vector of each person's ratings so they can be compared or plotted against each other.

Define x and y variables — 3

States the need to choose which variable goes on each axis (e.g., movie as x and rating as y) to examine relationships.

Student responses

1. clean your data for only people who have rated barbie and oppenheimer
2. collect all of the same movies into one row and then collect each rating into the corresponding column
3. Collect all the user ids and the scores for each movie and then create a y axis with their barbie rating and an x axis with their oppenheimer rating using the select method with their userids and then create the scatter plot.
4. create a table with user id, and barbie rating and oppenheimer rating and plot. join ratings with movies using the id, then filter for these two specific movies
5. filter by movie id that corresponds to babrbie and oppenheirmer then select rating for each individual user id
6. finding the movie_id of these two movies, matching people who rated these two, and comparing them as well as recording the numbers
7. First, I am going to find the corresponding movie_id of both movies from the movies table, then I will filter out the people who rate both movies from the ratings table.

Student responses

1. First, using the movies table, find the movie_id for Barbie and Oppenheimer. Then, discard any rows from the ratings table that do not include either of those 2 ids. Lastly, combine the two ratings that belong to the same user.
2. First, you would have to find the movie_ids of Barbie and Oppenheimer. Then, you would have to filter the ratings set by users who have ratings for both movies. Finally, you could mutate the set and create new columns that show the ratings for those two movies
3. For each user, connnect the movie id with the rating, can have
4. I need to distinguish people with each other, choose x and y to each represent a variable, whether rating for A or another
5. I think we need to group by movoies to better understand ratings in relation to them. Individual observatioons do not work.
6. I would first match the movie IDs to the movie titles so I can find the Barbie and Oppenheimer ratings. Then I would put each person's Barbie rating and Oppenheimer rating in the same row and use those two columns to make the scatter plot.

Student responses

1. I would need to be able to attach each rating for each film back to one user ID. If we are explicitly making the scatterplot for Barbie vs. Oppenheimer, then we need the move_id's for Barbie and Oppenheimer.
2. identify the two movies' id and keep the ratings of those two movies, put them in the same row for all the users
3. My first step would be to find the movie_id of Barbie and Oppenheimer. I would then find all the user_id's for those who have rated both Oppenheimer and Barbie, and create a new table that has the user_id, the barbie rating, and the oppehnheimer rating.
4. select all users who watched 2 movies and combine the tables so they are in the same place as the ratings
5. To make the scatter, we would need to isolate the ratings list by the two movies of interest, where each row is a user and their ratings for the two movies are the columns, then plot a scatterplot where each user is a dot, and each axis is a movie's rating.
6. We need to find the movie ids for Barbie and Oppenheimer, and then find every single user id who rated both Barbie and oppenheimer in the dataset.

Student responses

1. We need to match movie_id to response id based on similar user_ids
2. We need to search movies to find the movie_id for Barbie and Oppenhiemer. Then filter ratings to find people who rated both and filter the id and rating for each
3. We need to take all the user who rated barbie and Oppenheimer get their ratings
4. We will need to filter the ratings for Barbie and Oppenheimer, and then we'll need to create a table that displays the same user's ratings for both movies.
5. we would need the movie ID of Barbie and oppenheimer, then we would check a user rating of barbie and see if they also rated Oppenheimer, then pull that data into a table.
6. We would need to derive the survey response id for all these responses, and select only the two movies and each response's rating for those movies.
7. We would need to know the movie_ids of Barbie and Oppenheimer and then we would need to look at the ratings of those two movies amongst the same individual and their relationship
8. You would have to obtain a database of each rating of each movie.

Student responses

1. Filter out the rating on only Barbie and Oppenheimer. Makes a rating table data frame with user ids and a if else going through ratings to get their respective scores in the new data frame
2. filter the movies by the genres and plot the rating for each movie that fits the genre using the movie_id as a bridge for the rating set
3. filter the movies data to include only the movies titled barbie and oppenheimer to find their movie ids, then filter ratings data to include only people with ratings of both the two movie ids, then form a new table
4. filter the title to just get the two needed, then make a table with just those two titles and the reviews of them, alongside user id
5. Find the movie Ids for the desired movies and then filter out the ratings table to only include ratings for those movies
6. Group both movie ratings by id. That will give us a table and we can plot with each id being a point on the graph
7. I would begin by finding which movie ids correspond to Barbie and Oppenheimer, then create new variables that only include those IDs and their ratings by combining the data.

Student responses

1. i would filter ratings to only contain those that are rating the movie ID for barbie and oppenheimer. for these users i would have 2 columns, one for barbie nd one for oppenheimer. then i would plot both these ratings against each other for the table
2. I would need o be able to plot the selected data points on a graph relating the two, with rating on one axis, and movies on the other
3. Look at the movie_id column in movies, and look for the id of Barbie and Oppenheimer. After finding those ids, going back to the ratings table and filtering it down until it only has those ids, and then filtering that table to splitting the ratings by movie, and then creating the scatterplot form those ratings.
4. Need to figure out which is the independent variable and which is the dependent variable
5. Put each person's rating of both Barbie and Oppenheimer on a scatter plot. We need one axis to be a rating scale of Barbie and the other axis to be a rating scale of Oppenheimer. We need to ensure that each person is one point on the plot and not a duplicate.

Student responses

1. Search through the movie table to find the ids for Oppenheimer and Barbie. Filter the ratings sections for the two ids and group by response id.
2. select responses with movie id that matches barbie and oppenheimer and then create two columns with rows that match ratings for those movies
3. their barbie rating and their oppenheimer rating
4. using the movie IDs for barbie and Oppenheimer create a data set containing all of their reviews for all people, then filter out people who didn't rate both of the movies. Use that data to plot the points as (x, y) -> (Barbie, Oppenheimer).
5. We have to figure out what movie ID both Opponheimer and Barbie are. Then we have to filter the table to only show people's ratings for those two movies, and then we can start to do tests and comparisons on those ratings
6. We need to match the movie ID to each response ID and categorize the barbie movie and the openheimer movie to sort out each datapoint. If a user rate both movie, make sure to include both responses in the scatter plot.

Student responses

1. We would need to be able to select both the Oppenheimer and Barbie movies only from the movie plot. From the ratings plot, we would need to select the ratings for those movies specifically.
2. We would need to plot movie A's ratings on one axis and movie B's ratings on the other. Then this would show the correlation between two movies and what people have rated them. If we are trying to look at all the movies together, a bar graph owuld be better to visualize avergage rating
3. would need to filter for only barbie and oppenheimer reviews, join tables by user id to compare ratings of both movies by the same user
4. You want to find the movie ID for Barbie and Oppenheimer, then filter the ratings table by both those movie ID's and the user ID.
5. You would need to filter and select for people who have rated both movies and only select for those two movies, then set those two ratings to be your points. You need to define X and Y variables and plot!
6. You would need to get all the ratings for the movie ids of barbie and oppenheimer, then plot all the ratings on each axis

Student responses

1. You would want to create a new dataframe that contained the user's rating from the two movies in question. First determine the movie id of oppenheimer and barbie and then pull the ratings for those movies.

Student responses

1. Connect the tables on movie id.
2. I would join both data sets on movie_id and filter title to only barbie and oppenhiemer. Then I would create a table woth user_id and add a col with barbie and oppenhiemer movie rating
3. Join on movie id, then select only barbie and oppenheimer ratings
4. join table ratings with table movies, using a foreign key movie_id, then plot the paired ratings for each person to see the trend between the two ratings
5. join the tables on id
6. join the tables using the movie_id column to find how users rated specific movies by title
7. join them together on ids
8. To make the scatter, you would need to consolidate the two tables into one table, using the movie_id as the reference. The table would also include the user_id and rating columns, and filter so only the ids of Oppenheimer and Barbie are included in the scatter plot.
9. we would need to join ratings and movies by movie id

Student responses

1. You need to be able to merge the names/movie_id's, into a table that relates the two items to get a scatter.
2. You would need to come up with another quantitative value to group movies by (like another review or year)

Student responses

1. categorizing
2. Separate the date by ratings and movie name for each person by using the head code
3. We would need to know the various user_id's and ratings for the specific movies. I would then want to join these tables into one to show how user_id ratings are different for different movies, and then I would scatter two of the movies with all of the user id's for those two movies to identify a correlation
4. We would need to normalize the ratings if not yet normalized, and then sort them by movie/person so that they are all plotted correctly
5. We would need to use ggplot
6. Would need to know what the axes represent and how each movie_id would have its own rating. need to join the datasets so that the ratings from both movies can be compared in the same plot for each user
7. you would need some sort of ggplot, and then identify the x and y axis. i think there is a ggplot for scatter plots that would be able to plot this

Student responses

1. get the rate for those two movies for each one respond
2. I need to extract all ratings of the two targeted movies (search movies by id) and calculate the amount of each ratings.
3. In order to get the table that we want, the first thing we would need to do would be to pick out the two movie titles that we are looking for from the large base. We then want to match this up to movie id present in the people's rankings. Once we do this we are able to map onto how a specific user viewed a specific movie and then repeat the process for the other movie.
4. Knowing about how they rated for every film.
5. need data points
6. use the data function to clarify where youre pulling the data out of, use mutate() to make a table
7. You would need to start by grouping by movie id

Student responses

1. I think maybe you gotta use a random functino or something. Like something that will choose values at random in order to make it have that scattered behavior
2. plt? And set up x and y axis
3. Remove the na first
4. The ratings and their correlations with the movies based on each user
5. We need to get a plot with numbers on one side and the movies on the other, then in each row with the movie, have the ratings lined up on the plot.

Student responses

1. connect the ratings with the movies' name, choose the 2 movie you want to make plot with, retain only the movie name user_id and rating
2. Maybe create a value that is each persons ratings together in order to plot it to compare with other peoples answers.
3. We need to identify the movie we want to compare to

Student responses

1. I would label the axis of plots as movies and ratings
2. one movie as x-axis, one movie as y-axis
3. We need to define our x and y variables, x will be the movie and y will be the ratings and see if we can find a correlation between them

Which movies did this person rate, and how many stars did they give each? Use `filter()` on `movies` to look up each movie ID. Explain which column connects the two tables.

79 anonymous responses

Correct movie names and ratings — 46

States both Oppenheimer and Barbie were rated, with Oppenheimer 5 stars and Barbie 4.5 stars.

Correct ratings but no movies named — 6

Provides the two correct numeric ratings (5 and 4.5) without explicitly naming which movies they correspond to.

Movie names only or partial listing — 6

Mentions only the movie titles without providing ratings.

Technical lookup or id-focused answers — 6

References movie IDs or filter code to locate movies rather than stating names and ratings directly.

Response themes for question 3

Correct concise restatement Briefly states the two movies and their ratings (Oppenheimer 5, Barbie 4.5) in a short form.	5
Correct ratings plus join column noted Gives the same ratings and additionally names the joining column between tables (movie_id), with a small typo in rating format.	4
No answer or incomplete attempt Indicates the respondent did not complete the task or failed to find the answer in time.	4
Other responses	2

Student responses

1. Barbie and oppenheimer
2. Openheimer and Barbie
3. Ophienhimer recieved a 5 and bribie recieved a 4 and connect movie_id and rating
4. oppenheimer - 5, barbie - 4.5
5. Oppenheimer 5 stars and Barbie 4.5 stars. The movie_id connects both columns.
6. oppenheimer a 5 star and barbie a 4.5 star, the movie_id column connects the tables
7. oppenheimer and barbie
8. oppenheimer and barbie
9. oppenheimer and barbie

Student responses

1. oppenheimer and barbie
2. Oppenheimer and barbie
3. Oppenheimer and Barbie
4. Oppenheimer and Barbie
5. Oppenheimer and Barbie
6. Oppenheimer and Barbie
7. Oppenheimer and Barbie
8. Oppenheimer and Barbie
9. Oppenheimer and Barbie

Student responses

1. Oppenheimer and barbie, with 5 and 4.5 stars respectively, connected by the movie_id
2. Oppenheimer Barbie
3. oppenheimer first movie rated 5; barbie rated 4.5
4. Oppenheimer(5) and Barbie(4.5)
5. Oppenhimer 5 stars Barbie 4.5 stars
6. rated barbie (288513) a 4.5 oppenheimer (287699) a 5
7. The first row's movie is Oppenheimer. The second row's movie is Barbie. The column that connects the two table is movie_id
8. The movies are Barbie and Oppenheimer
9. These are Oppenheimer and Barbie, oppenheimer got 5 stars and barbie got 4.5 stars

Student responses

1. These movies are Barbie and Oppenheimer, this use rated Oppenheimer a 5 and Barbie a 4.5. These two tables are connected on movie_id.
2. They gave ophen a 5 and barbie a 4.5
3. They rated barbie and oppenheimer, giving opp a 5, and barbie a 4.5
4. they rated Openheimer 5 and Barbie 4.5 stars
5. They rated Oppenheimer 5 and Barbie 4.5
6. This is Oppenheimer and Barbie
7. This person gave Oppenheimer 5 stars, and Barbie 4.5 stars. The movie_id column connects the two tables.
8. This person rated 2 mvoies. THey rated them 5 stars and 4.5 stars. Oppenheimer and Barbie were the movies
9. This person rated Barbie with a 4.5 stars and Oppenheimer with 5 stars

Student responses

1. This person rated both Oppenheimer and Barbie. The person gave Oppenheimer 5 stars and Barbie 4.5 stars. The column that connects the two tables is the user_id.
2. This person rated Oppenhaimer at 5 and Barbies at 4.5.
3. This person rated Oppenheimer (2023) and Barbie (2023). They rated Oppenheimer 5 stars and Barbie 4.5 stars.
4. This person rated oppenheimer (5) and barbie (4.5). the movie id connects the two tables
5. This person rated oppenheimer 5 stars and barbie 4.5 stars.
6. This person rated Oppenheimer 5 stars and Barbie 4.5 stars.
7. this person rated Oppenheimer and Barbie.
8. This person rated Oppenheimer and Barbie. They gave Oppenheimer a 5, and they gave Barbie a 4.5. movie_id connects the two tables.
9. This person rated oppenhiemer and barbie. And he gave them 5 and 4.5 starts respectivley

Student responses

1. this user rated Oppenheimer 5 stars and Barbie 4.5 stars. the movie_id column connects the two tables.

Student responses

1. one is 5 one is 4.5
2. The give 5 and 4.5 stars user id connects two table
3. they gave 5 and 4.5 stars
4. They rated 5 and 4.5 stars, and you can find the movie titles by linking movie ID
5. They rated a 5 and a 4.5 for the two movies.
6. They rated one movie 5, and one movie 4.5.

Student responses

1. 4.5 stars Barbie and 5 stars Oppenheimer, connected on movie_id
2. Barbie and Oppenheimer
3. Oppenheimer & barbie
4. Oppenheimer and Barbie
5. The movie_id is the key column. Barbie 4.5 and Oppenheimer 5
6. They rated Oppenheimer and Barbie.

Student responses

1. 288513 Barbie (2023) 2023 Comedy;287699 Oppenheimer (2023) 2023 Drama
2. I used the filter(movie_id == __) command to find out that the titles are "Oppenheimer" and "Barbie".
3. movies |> filter(movie_id == 287699 | movie_id == 288513) oppen, barbie The movie_id column connects them
4. Oppenheimer. filter(movie_id=="287699"
5. The person rated movies with id 287699 and 288513 filter(movies$movie_id)
6. use the filter(movie_id == "287699", "288513")

Student responses

1. barbie 4.5
2. I think the rating score is 5 and 4.5, and the movie_id is 287669, and 288513
3. Oppenheimer (2023)
4. Oppenheimer is one of the ids and the second is barbie
5. Oppenheimer, 5, 4.5

QUESTION 3 · Correct ratings plus join column noted · 1/1

Student responses

1. movie_id works as key to connect them, where one represents Barbie(2023), and the other movie that we are considering. The person gave Barbie 4.5 and the other one 5
2. This person gave Oppenheimer a 5, and Barbie a 4. movie_id connects the two tables.
3. This person rated barbie and oppenheimer, giving the former 5 stars and the later 4.5.
4. User 874 rated 5 stars and 4.5 stars. The movie_id column connects the ratings table to the movies table.

Student responses

1. i cant find it this is too little time
2. I did not have enough time to figure it out
3. i didn't find it in time to answr
4. oppenheimer

How would you build that table using the tools we've learned? Describe your steps in words. How will you make sure the two ratings in each row belong to the same person?

88 anonymous responses

Use a common ID to match rows — 26

Suggests using the same participant ID to ensure ratings in a row belong to the same person and then display or collect that person's responses.

Use a join operation — 21

Indicates combining tables or columns by performing a join to align ratings from the same person.

Use mutate to create or adjust columns — 19

Proposes transforming the table by mutating columns (creating or modifying variables) to build the desired table structure.

Group by person or title to align ratings — 7

Advocates grouping rows by person (or by title then aligning via user id) to produce rows that contain both ratings for the same individual.

Response themes for question 4

Speculative or unsure approach — 6

Responses that are tentative, vague, or indicate uncertainty about the method to use.

Create a new column to link ratings — 3

Recommends creating an additional column to store or link ratings so they can be associated per person.

Filter rows by user id — 3

Describes selecting or filtering rows to show only records belonging to a specific user id so both ratings appear together.

Flatten table to wide format — 3

Suggests reshaping or flattening the data so each rating type becomes its own column, producing one row per person with separate rating columns.

Student responses

1. check the user id and then get their movie rating
2. Collect the same ID number response and display out all of their responses. Remove the movie names and strictly display out their rating. We also need to filter and joining tables of the user_id
3. Filter for the movie_ids we want then pair them by user_id
4. filter the row base. and Use the common ID to match it
5. for each user id join the data by mutating the data and creating two new columns for barbie rating and Oppenheimer rating per user
6. group by user_id and filter for both movie ratings
7. Have the same person's ratings be in the same row for the two movies
8. I think we need to left join the table on itself by user id, so both ratings are in the same row, and then just drop the movie titles
9. I would get the Barbie and Oppenheimer ratings separately, then join them using user_id. The user_id makes sure that the two ratings in each row are from the same person.

Student responses

1. I would make separate columns for the movie ratings of interest, named barbie_rating and oppenheimer_rating, then rearrange the table by user id so that each row was one user id, with their value in the specific rating columns being located through their user id, and the movie's movie id.
2. I would name both the rating columns barbie_id and opperheimer_id and then use the join feature to join both of these columns
3. I would use the table and filter method to make sure it's the same person
4. I would use the user_id and left join so that their ratings pop up. I think selecting each rating into one row would work.
5. MATCHING THEIR USER ID
6. To build that table, you have to use the "left join" command to combine multiple data sets. Also check data so it doesn't belong to multiple people.
7. To make sure they belong to the same person you can match up each movies' rating with user_id, and you essentially need to flip the table.
8. u can left join by both user id and selected rating and categorised by the two movies

Student responses

1. use the same user id to find rating for film1 and rating for film 2, then join these tables using foreign key user_id
2. user_id join, one row 2 ratings
3. We can create a new column that holds the ratings for the 2nd movie, so a row would contain the user id, the rating for Barbie, and the rating for Oppenheimer. In that case, the matching user id would be the way to verify that the ratings were provided by the same person.
4. We should connect them by using the key user_id
5. we should have each column of the table be a movie, the row will the user id and then their rating for the film of the column
6. You can group by user_id and the filter by user_id's that have given ratings to both movies
7. you could sort by id of the submitter to ensure each row belongs to the same person
8. You join the two and filter out the same id
9. You would build a table that is one row per user id where one column is the barbie rating one is the oppenheimer rating, you can filter by id

Student responses

1. build seperate oppenheimer and barbie tables that have user_id and ratings columns then join those tables using user_id to create the final table
2. Build seperate tables with the ratings, and then join them to create rows with one user's ratings for both movies.
3. Create two separate tables and join them using user_id. Then filter by user_id
4. Filter movie name to keep Barbie and Oppenheimer, Group by title, then join on user id, keeping the barbie and Oppenheimer ratings as separate columns
5. I would make one table with only the Barbie ratings and another with only the Oppenheimer ratings. Then I would join the two tables using user_id so each row has both ratings from the same person.
6. I would merge the two rows if and only if they have a rating for both Barbie and Oppenheimer, and have it include the Barbie rating, the Oppenheimer rating, and join it all with the user id
7. i would use a join by person_id to make sure it all maps from the same person.
8. I would use a join to merge the ratings and movies and then filter to only include Barbie and Oppenheimer. I would then group by user

Student responses

1. I would use join to make sure every user ID is properly linked to the movie_ids
2. I would use the filter option in combination with left join?
3. join
4. Join by user id
5. Join the two tables by user_id
6. join the user id to there barbie and openheimer score
7. joins
8. Select person_id, rating, movie_id and group by refer to same person.
9. using join operation

Student responses

1. We could use left join and try to match them by user_id into one table
2. we could use left join to match them by user rating and then display both ratings
3. We should join each rating with regards to a key in people ID
4. You use the join function to join the ratings of the two movies from each user_id together

Student responses

1. Add columns to data that include each users rating for barbie and Oppenheimer based on looking at each user_ids ratings for those movies.
2. filter the data to include one of the movies then mutate the ratings to a new column called "movie_rating" then repeat witht the other movie
3. group by userid, then mutate by movie title to have two columns with each rating.
4. i would create a new table and mutate so that barbie rating is one col and oppenheimer is another col
5. I would create new columns for barbie and oppenheimer ratings using mutate, and assign the variables to the ratings of each title. I would also group by user ID.
6. I would create new variables for oppenheimer and barbie and apply the ratings from their respective rows into the new variable for each individual user.
7. I would create two new columns, barbie_rating and oppenheimer_rating, using mutate, where the value is given for the column if it matches the movie_id for the appropriate movie. To make sure the ratings belong to the same person, there will be a requirement for the user_id to match for both.
8. I would mutate the columns.

Student responses

1. i would mutate the table to add 2 columns, a barbie and an oppenheimer rating. the barbie one would take that users rating for barbie and the oppenheimer would take the same for its oppenheimer rating. then select just those columns to analyze
2. I would mutate to make new columns for both barbie rating and oppenheimer rating. Then we would need to sort each rating based on their title to get the right ratings with the right movie.
3. I would use mutate to create two new columns that would include both of the wanted ratings. Then, I would use select to include just the id of the person and the two new columns
4. Mutate a column with each persons Barbie rating and do the same with each person's Oppenheimer ratings. Select for them in a new table.
5. Use a common Id to match rows, and then use mutate to add the score of Barbie and Oppenheimer.
6. Use mutate to add a column based on movie rating assigned to each movie
7. Use mutate, filter and then group by
8. we could use the mutate function to create two new columns that use ifelse statements to fill out the values of the ratings for the two movies

Student responses

1. We should make a new table where we make a column users by pulling user id and then there ratings using an if else
2. We would probably mutate a new col with barbie and oppenhiemer ratings and group by user_id
3. You would need to make sure that you are grouping by user id so that each user will only respond to one row at the most. You also now need to create new columns for barbie and oppenheimer which you can then find a way to link to the specific rating that was given for that movie.

Student responses

1. create new columnns for barbie and oppenheimer ratings that filter for those ratings only for users that rated both
2. filter by ratings, then separate between the two users using name and id,
3. group by respondent
4. group by user id and find a way to transpose the day, maybe mutate and make a new row for barbie and openhiemer
5. Maybe we could group by person for the rows.
6. table() check the id of two people
7. You can use group_by(user_id)

Student responses

1. Maybe there is a way to join the variables in a table, like maybe the $variable logic might be the way to accomplish this task, but this is just me thinking out loud
2. Mutate? or summerize by person
3. S
4. uncertain
5. use conditional validation to confirm that the rating id matches for each review
6. you can select which headings you want in the top row of the table and then filter the responses

Student responses

1. creat another column
2. create variables called barbie rating and oppenheimer rating and then you can select those variables in a single row with user id
3. we could make one user's rating horizontal first and change the rating to each movie name rating

Student responses

1. First filter all ratings with only the two films.
2. I think we should create a table for named_rating first, and having three columns about their rating and
3. I would filter user_id rows that contain Openheimer and Barbie movies, and then join them togetehr based on common variable (user_id)

Student responses

1. We need to flatten the table, creating new columns for each rating type of the movies.
2. You could do this by somehow merging the "movie title" and "movie rating" columns into columns that simply give the ratings for the two movies
3. You would have to create separate columns for the ratings and then perform a join while filtering so that we get the ratings for each desired title/movie_id

Change the movie title and the rating column's new name. Submit your two changed lines. Why should we keep user_id?

80 anonymous responses

Select user_id and rename rating Proposes code that filters for Oppenheimer and selects user_id plus a renamed rating column (e.g., oppenheimer_rating or opp_rating).	46
Emphasize join key between tables Explicitly explains keeping user_id because it is needed to join Oppenheimer and Barbie ratings or combine tables.	13
Unique identifier / tracking rationale States user_id is necessary as a unique identifier to track users, ensure uniqueness, or know which user gave which rating.	7
Minor wording or casing variants Short or variant attempts that restate the filter/rename with small differences in casing, spacing, or punctuation.	6

Response themes for question 5

Keep user_id—identify same rater States we should keep user_id so ratings can be tied to the same person, enabling identification or matching across movies.	3
Typo or swapped title/rating Contains a misspelled title or mistakenly renames rating to a different movie's column, indicating confusion or error.	3
Other responses	2

Student responses

1. # Change this example to find Oppenheimer's ratings. named_ratings |> filter(title == "Oppenheimer (2023)") |> select(user_id, opp_rating = rating) |> head(4)
2. # Change this example to find Oppenheimer's ratings. named_ratings |> filter(title == "Oppenheimer (2023)") |> select(user_id, oppenheimer_rating = rating) |> head(4). We want to keep the user_id so that we can join on it
3. # Change this example to find Oppenheimer's ratings. named_ratings |> filter(title == "oppenheimer") |> select(user_id, oppenheimer_rating = rating) |> head(4)
4. # Change this example to find Oppenheimer's ratings. named_ratings |> filter(title == "Oppenheimer") |> select(user_id, oppenheimer_rating = rating) |> head(4) We keep user ID because we want to see the same user's ratings to both movies
5. Because the user_id is a unique identifier, and the other things wont change it
6. Because we want to make sure the rates belong to the same person
7. Change title to == "Oppenheimer (2023)" and the barbie_rating to oppenheimer_rating. It should be the same user_id since they are the same person rating the two movies.

Student responses

1. Changed the Barbie words to Oppenheimer. You keep User_id because it will be the connecting column when joining the two tables
2. filter(title == "open Hemiler"") |> select(user_id, open_hemiler_rating = rating) |>
3. filter(title == "Oppenheimer (2023)") |> select(user_id, oppenheimer_rating = rating) |>
4. filter(title == "Oppenheimer (2023)") |> select(user_id, oppenheimer_rating = rating) |>
5. filter(title == "oppenheimer (2023)") |> select(user_id, oppenheimer_rating = rating) |> head(4) because the user_id is for all the people
6. filter(title == "Oppenheimer (2023)") |> select(user_id, oppenheimer_rating = rating) |> head(4) we need to keep user_id because it will allows us to see which people have rated oppenheimer
7. filter(title == "Oppenheimer (2023)") |> select(user_id, oppenheimer_rating = rating) |> keep user id to join the ratings
8. filter(title == "Oppenheimer (2023)") |> select(user_id, oppenheimer_rating = rating) |> User id ensures unique ratings
9. filter(title == "Oppenheimer (2023)") |> select(user_id, oppenheimer_rating = rating) |> we keep the user_id as the key to join two tables

Student responses

1. filter(title == "Oppenheimer (2023)") |> select(user_id, oppenheimer_rating = rating) |> we keep user id so that we can map the two ratings made by the same person together in the new combined table.
2. filter(title == "Oppenheimer (2023)") |> select(user_id, oppenheimer_rating = rating) |> We keep user_id so we can match each person's Oppenheimer rating with the same person's Barbie rating
3. filter(title == "Oppenheimer (2023)") |> select(user_id, oppenheimer_rating = rating) |> we need to keep user_id because that is what we are using to join the tables. the tables are joined based on the user_id
4. filter(title == "Oppenheimer (2023)") |> select(user_id, oppenheimer_rating = rating) |> we should keep user_id because we want this to be included to join the tables together once we are done
5. filter(title == "Oppenheimer (2023)") |> select(user_id, oppenheimer_rating = rating) |>. So that we can group the two movies ratings together

Student responses

1. filter(title == "Oppenheimer (2023)") |> select(user_id, oppenheimer_rating = rating) keep user_id so we can tell which user gave which rating, which will help when we join this with the barbie rating.
2. filter(title == "Oppenheimer (2023)") |> select(user_id, oppenheimer_rating = rating) keep user_id to match Barbie and Oppenheimer ratings to the same person
3. filter(title == "Oppenheimer (2023)") |> select(user_id, oppenheimer_rating = rating), that is going to be our primary key so we need to keep it
4. filter(title == "Oppenheimer (2023)") |> select(user_id, oppenheimer_rating = rating), We should keep user_id so we can match each person's Oppenheimer rating with their Barbie rating later.
5. filter(title == "oppenheimer") |> select(user_id, oppenheimer_rating = rating) |>
6. filter(title == "oppenheimer") |> select(user_id, oppenheimer_rating = rating) we should keep user id because that is the key we will join on

Student responses

1. I would change lines 3 and 4 to filter(title == "Oppenheimer (2023)") |> and select(user_id, oppenheimer_rating = rating) |>. We should keep "user_id" because we need to attribute the correct ratings to each individual response via join.
2. named_ratings |> filter(title == "Oppenheimer (2023)") |> select(user_id, oppenheimer_rating = rating) |> head(4)
3. named_ratings |> filter(title == "Oppenheimer (2023)") |> select(user_id, oppenheimer_rating = rating) |> head(4) The user id will be the same for both because the same user rated each movie
4. named_ratings |> filter(title == "Oppenheimer (2023)") |> select(user_id, oppenheimer_rating = rating) |> head(4) We are basically sorting based on user_id so getting rid of it would do us zero good
5. named_ratings |> filter(title == "Oppenheimer (2023)") |> select(user_id, oppenheimer_rating = rating) |> head(4) We keep user_id because we need that column to join this table with the Barbie table.

Student responses

1. named_ratings |> filter(title == "Oppenheimer (2023)") |> select(user_id, oppenheimer_rating = rating) |> head(4) we should keep user_id because that is how we will join the two tables
2. named_ratings |> filter(title == "Oppenheimer (2023)") |> select(user_id, oppenheimer_rating = rating) |> head(4) we want to keep user_id, so that we know who gave each rating, and so that we can join the two tables using user_id to make the final table.
3. named_ratings |> filter(title == "Oppenheimer (2023)") |> select(user_id, oppenheimer_rating = rating) |> head(4). keep user_id to know which person's rating
4. named_ratings |> filter(title == "Oppenheimer (2023)") |> select(user_id, oppenheimer_rating = rating), have to keep user_id to match a person with their ranking
5. named_ratings |> filter(title == "Oppenheimer (2023)") |> select(user_id, opponheimer_rating = rating) |> head(4) Keep user id because that is how we join the tables
6. oppenheimer -> named_ratings |> filter (title=="Oppenheimer (year)" |> select (user_id, oppenheimer_rating = rating) |> head (4)

Student responses

1. Oppenheimer <- named_ratings |> filter(title == "Oppenheimer (2023)") |> select(user_id, Oppenheimer_rating = rating) |> head(4) Keep User id because irt allows us to join the table sin future
2. oppenheimer <- named_ratings |> filter(title == "Oppenheimer (2023)") |> select(user_id, oppenheimer_rating = rating) oppenheimer |> head(4) we should keep user ID so that we can match it up with the same person's rating for barbie.
3. REPLACE BARBIE WITH OPPENHEIMER
4. select(user_id)
5. So we can join the two ratings together by user ID, linking each person's specific ratings.
6. title == "Oppenheimer (2023)" select(user_id, oppenheimer_rating = rating)
7. We need to keep user_id in order to join the two tables later.
8. We should keep user_id so we can use the join function later.

Student responses

1. Because this would include all the relevant info that the dataset needs while ikeeping it concise and optimized, the user id is a unique identifier
2. filter(title == "Oppenheimer (2023)") |> select(user_id, Oppenheimer_rating = rating) |>
3. filter(title == "Oppenheimer (2023)") |> select(user_id, oppenheimer_rating = rating) |> Keep user ID because it will help us join with barbie_rating
4. filter(title == "Oppenheimer (2023)") |> select(user_id, oppenheimer_rating = rating) |> user_id is the key that helps us connect the Barbie movie and the Oppenheimer
5. filter(title == "Oppenheimer (2023)") |> select(user_id, oppenheimer_rating = rating) |> we need it to join the tables
6. filter(title == "Oppenheimer (2023)") |> select(user_id, oppenheimer_rating = rating) |> We should keep user_id because it is how we can keep track of our initial responses.
7. filter(title == "Oppenheimer (2023)") |> select(user_id, oppenheimer_rating = rating) |> we should keep user_id because we need to join the ratings for oppenheimer and barbie on the user_id so that every row represents an individual.

Student responses

1. filter(title == "Oppenheimer (2023)") |> select(user_id, oppenheimer_rating = rating), We should keep user_id to track that someone didn't rate the movie twice
2. I changed the filter and select lines into filter(title == "Oppenheimer (2023)") and select(user_id, oppenheimer_rating = rating). That user_id should be kept is because we'll need to use it to join both tables in the future.
3. named_ratings |> filter(title == "Oppenheimer (2023)") |> select(user_id, Oppenheimer_rating = rating) |> head(20) We should keep user ID for now so it includes users who recorded barie or
4. oppenheimer <- named_ratings |> filter(title == "Oppenheimer (2023) |> select(user_id, oppenheimer_rating = rating)
5. User id is important to because it tells us who rated each
6. we should keep user_id to be able to match the same user's barbie and oppenheimer rating together.

Student responses

1. Because we still need the users and how they rated oppenheimer
2. fo runiqueness
3. It is our unique identifier for the table
4. selected(user_id) and rename
5. to keep them together
6. to match
7. We want to compare when there is the same user, what their rating towrds different movies, to make a comparsion

Student responses

1. change it to ppenheimer
2. filter(title == "oppenheimer") |> select(user_id, oppenheimer_rating = rating) |>
3. name are the same
4. so we can maintain the users id
5. title == "oppenheimer(2023)"
6. You first need to change the title to be Oppenheimer. Then you need to change barbie_Rating to oppenheimer_rating

Student responses

1. Keeping user id makes sure each response and pair of rating can be identified as unique.
2. we should keep user id so we know the rating comes from the same person
3. We should keep user_id because it is what is connecting and providing the relationship between the ratings of a user for the Oppenheimer and Barbie movies.

Student responses

1. named_ratings |> filter(title == "Oppenhaimer") |> select(user_id, barbie_rating = rating) |> head(4)
2. Openheimer <- named_ratings |> filter(title == "Openheimer (2023)") |> select(user_id, openheimer_rating = rating) |> head(4)
3. oppenheimer <- named_ratings |> filter(title == "Oppenheimer (2023)") |> select(user_id, oppenheimer_rating = rating) |>

Which Oppenheimer rating belongs beside each Barbie row, in order? Explain how R should find the matches when the rows are in different orders.

0 anonymous responses

No responses were submitted.

What do we know about these people's ratings, and what don't we know? Would replacing the missing values with zero be reasonable? Explain.

89 anonymous responses

Rated Barbie but not Oppenheimer — 45

Responses indicating the available information: the person provided a rating for Barbie but did not provide a rating for Oppenheimer (missing for Oppenheimer).

Zero imputation misrepresents data — 13

States that replacing missing ratings with zero is inappropriate because it would introduce false information or be unhelpful.

Zero imputation skews statistics — 12

Notes that replacing missing values with zero would change or skew summary statistics and should be avoided.

Missing due to not having seen film — 7

Argues missing ratings likely mean the person hasn't seen Oppenheimer, so imputing zero (as a negative preference) is inappropriate.

Response themes for question 7

NA means nothing about opinion Emphasizes that NA conveys no info about the user's opinion and so cannot be interpreted as a low rating.	5
Other responses	4
Prefer drop rows or alternative imputation Suggests removing rows with missing Oppenheimer ratings or using mean/median/inner join instead of imputing zeros.	3

Student responses

1. N/A indicates that there is no data available for that person. Putting a 0 in place would then assign it a numeric value and there could still get plotted on the scatter plot and would add to the data when really there is no data.
2. No, zero wouldn't be the same because they didn't rate the movie at all and perhaps haven't watched it. We know that these users only rated barbie. They still may have liked oppenheimer.
3. The n/as signify people that rated Barbie but not Oppenheimer. Replacing the missing values with zero would unfairly skew the data, so we should not do this. Instead, we should just remove these users' ratings.
4. The NAs came from someone rating the barbie movie but not rating the oppenheimer movie. Replacing values with 0 is NOT a good idea because it would be inputted into our data that the rating given was 0, which would alter our data.
5. The NAs come from the users not rating all of the other movies. Replacing them with zero could be a problem because the average rating would be lower

Student responses

1. these people ranked barbie but not oppenheimer, we don't know their exact ranking so replacing the NA with 0 would be inaccurate and drag down our correlation
2. These people rated barbie and not oppenheimer. Replacing the values with zero would not be viable because ot would bias the data towards higher barbie reviews and not give an accurate comparison. Zero is also out of the rating scale.
3. These people rated barbie but not oppenheimer. It would not be reasonable to replace n/a with zero because a 0/5 rating isn't the same as not watching the movie
4. These people rated barbie but not Oppenheimer. replacing the values with 0 would be really bad because then we'd have a ton of data of people who watched only barbie. it would decrease oppenheimer's average rating and it wouldnt make sense.
5. They did not rate Oppenheimer, but did rate Barbie. Cant replace with zero because they just didnt rate it, so we ignore it.

Student responses

1. We know how they rated Barbie but not how they might have rated Oppenheimer. Replacing the NAs with 0 would not be reasonable as we don't know that they would actually rate the movie poorly
2. We know if the people rated both the movies from their rating columns. No, it would not be reasonable to replace the missing values with 0 because that would highly influence our conclusion
3. We know that not everyone rated both movies. And no we should not replace the NAs woth 0s because that is innaccurate data
4. We know that these people gave a rating for the Barbie rating, but did not give a rating for the Oppenheimer rating. We cannot show a correlation between users liking these two movies from the NA response. If you replace with a zero, it will incorrectly say that the users rated Oppenheimer 0 stars.
5. we know that these people have barbie but not oppenheimer ratings, don't replace the na with 0
6. We know that these people rated barbie but did not rate oppenheimer. Replacing the missing values with zero would not be reasonable as it would mess up any further analysis we want to do after.

Student responses

1. we know that these users rated barbie but not oppenheimer. it is not reasonable to replace the na with zero because they just havent watches/rated the movie
2. we know that they all rated barbie but not oppenheimer. We don't know their ratings to oppenheimer so we shouldn't replace them with 0. This would create biased effect.
3. We know that they did not have an opinion on the oppenheimer movie. Replacing them is a bad idea since this would skew the data in the lesser direction, so you shouldnt replace them since that adds bias
4. We know that they don't have a rating for oppenheimer, don't replace it with zeros because then you'd be giving them a zero rating rather than no rating
5. We know that they gave barbie a rating but we don't know how they feel about oppenheimer. we should not replace NA with zeros because that would falsely give oppenheimer low ratings
6. we know that they only rated barbie, we don't know if they have seen both movies, or if they like barbie more than oppenheimer, you should not replace the missing values with zero because that would add data that isnt real.

Student responses

1. We know that they rated barbie but not oppenheimer. If zero were replaced for NA value it can mess up further caluclations based off of this new table
2. we know that they rated barbie, but we don't know if they rated oppenhiemer. since it is likely that they did not rate oppenhiemer at all, we should not replace it
3. We know that they rated Barbie, but we don't know what they rated oppenheimer. Replacing the missing values with zeros would misrepresent the data. The didn't give Oppenheimer 0 stars, they just didn't rate it.
4. We know that when there is a NA, that person didn't rate the movie. We don't know if they enjoyed the movie or not though. We shouldn't replace it with a 0, because that would skew the data, leading to results that aren't necessarily true.
5. We know the barbie_rating but we do not know these people's oppenheimer_rating. I would not replace these missing values with 0, that will skew the data incorrectly

Student responses

1. We know their barbie rating but not their oppenheimer rating. This means we just dont know their rating but we have no way of figuring out how to predict their ratings before we analyze the data more
2. we know their barbie rating but we dont know for oppenheimer, so they might not have watched//rated oppenheimer. replacing it with 0 would be bad bc it would assume they all disliked oppenheimer. instead we should remove this data to only keep those who have rated both movies
3. We know their barbie ratings but not their oppenheimer rankings. We should not replace the missing values with zero because that would skew the oppenheimer data.
4. we know their barbie ratings but not their oppenheimer ratings. putting a zero for the ratings we dont know would not be okay because that could skew the data lower than it actually is.

Student responses

1. We know their rating for the first movie, but not their rating for oppenheimer. Replacing these values with zero wouldn't be reasonable because that just means that they gave oppenheimer a zero rating, which is false, and it would affect whatever data analysis processes we would be carrying furtuer.
2. we know their ratings for barbie but not Oppenheimer. it would not b reasonable as 0 can be a valid rating and it would be skewing the data with an erroneous report
3. We know their ratings for Barbie, but we don't know their ratings for Oppenheimer. It would be unreasonable to replace those missing values with zero since we would otherwise interpret results as they hated the movie and gave 0 points to Oppenheimer.
4. We know these people rated Barbie, but we do not know their Oppenheimer rating because it is missing. Replacing the NAs with 0 would not be reasonable because 0 would mean they gave Oppenheimer a zero rating, when really we just don’t have a rating from them.

Student responses

1. We know these people rated Barbie, but we do not know their Oppenheimer ratings. The missing values should not be replaced with 0 because NA means we do not have a rating, while 0 would mean they actually gave the movie a rating of zero.
2. We know these peoples Barbie ratings but we don't know their Oppenheimer ratings. Replacing NA with 0 would be very incorrect because it would set their ratings to 0/5 stars which would completely mess up the data.
3. We know they rated Barbie, but not Oppenheimer. It would not be ok to replace with 0 since we don't know what they would rate it as.
4. We know they rated barbie, but we don't know if they rated oppenheimer.
5. We know they watched Barbie, and their ratings on the movie. We don't know their ratings on Oppenheimer. Replacing these values with 0 would not be appropriate, as that would skew the data towards lower ratings for Oppenheimer.
6. We know those people's id the rating of barbie. But change the NA to 0 is a bag idea

Student responses

1. We know what their ratings for Barbie are, but not for Oppenheimer. Replacing the missing values with zeros would not be reasonable, because that implies they gave Oppenheimer zero stars. The should just simply be excluded in the scatterplot.
2. We know what they rated and what they did not. An NA value means they did not rate the movie so we can not safely assume their rating without messing up our data.
3. We know what they rated Barbie, we don't know what they rated Oppenheimer. 0 would not be correct because it would unfairly bring down the average rating
4. We know which movies they've rating and what that rating was, but we don't know if these values are correlated. Replacing the missing values with zero would not help because it is incorrect: the user did not rate the N/A movie zero stars, they simply didn't rate it at all.

Student responses

1. no because that is false information and not what we looking for
2. No it would not be reasonable to replace msisign values with zero because then it is dropping their rating down to the worst possible value. In reality, they might have enjoyed the movie, but they might have not rated the movie. This would have a really bad affect on the correlation.
3. no it would skew the NAs to be much lower ratings when calculating
4. No, because we don't know what they actually think of the movie, assuming zero would destroy the dataset, and they should instead be filtered out.
5. no, it will affect distritbution
6. Replacing the missing value with a 0 would not help
7. Replacing the missing values with zero would not be reasonable because that would imply that they hated the movie and gave it zero out of five stars

Student responses

1. Replacing with zeroes would not be reasonable; people who are attracted to Oppenheimer are likely not the correct audience to be watching Barbie, thus it's likely that they'll rate it a little lower, but not so much lower as to get that sort of response. Instead, I'd replace them with the mean value of Oppenheimer watches who watched Barbie, but I'd rather just drop those rows.
2. they didn't rate the movie. replacing it with a zero would not make sense
3. we cannot replace with zero as it simply is a missing input and its theoretical value is not zero. you should instead eliminate the inputs with a zero
4. We do not know their oppenheimer rating, and replacing them with 0 is dangerous, could likely go with the median or something like that
5. We do not know what the people thought about Oppenheimer, whether they watched it or not, or whether they just did not rate it. Replacing with 0 would add improper noise that is not necessarily reflective of what they thought/would/will think of the movie.

Student responses

1. We know these people rated barbie but not oppenheimer, a zero would be reasonable if we are assuming they didn't like oppenheimer and felt it wasn't necessary to rate it, however this isn't implied

Student responses

1. No because that would affect the ratings negatively, where NA wouldn't affect the mean ratings or other stats
2. no because that would change the statistics. They should just be dropped
3. No because that would decrease the overall ratings of that particular movie, this means that for a missing response, replacing an 0 means they would actually rank it super low when in reality they just didnt response
4. no because that would severly distort the ratings yo
5. No, because that would skew the ratings
6. replace with 0 is not reasonable it creates outliers
7. These NAs come from people who rated Barbie but did not rate Oppenheimer, replacing with 0 would not be reasonable as that would drastically skew the data.
8. They have not rated openheimer. We should not just replace the missing values with 0 since that might skew the data
9. We don't know if they watched it or not. We cannot replace with 0 because it would drag down the average rating.

Student responses

1. we know that these are people who rated barbie but not oppenheimer. Replacing the missing values with zero would not be reasonable because it would severely tank the data
2. we know that they haven't rated Oppenheimer. we shouldn't replace with zero because we don't know that that's what they'd write it they had watched it
3. We know they rate for barbie, but not the other film. Not reseaonable because 0 stands for the lowest point. We should just drop these people or use average

Student responses

1. missing due to not having seen film
2. no because they most likckly just havent seen oppenheimer not that would hate oppenheimer. because it would lead to peple who have not had an opionion on oppeheimer would lead to them having a negitive opionion compared to no opionion
3. Some people didn't watchopenheimer , no replacing with zero is not reasonable as that wild rage overall rating tdown when they mgiht ahv enejoyed the movei
4. Their Barbie ratings were 4's and 3's, but they didn't see Oppenheimer. These ratings shouldn't be replaced with 0 because it would skew results lower.
5. These people didnt rate oppenheimer replacing them with 0 would not make sense as they might not have reviewed it due to disliking it. 0 would also distort the avg for the oppenheimer data
6. they did not rate the oppenheimer movie, so we dont know how they feel. giving them 0 would skew the data
7. they probably havent watched the movie. you cant replace it with 0 in this case beacue it would heavily skew the data

QUESTION 7 · NA means nothing about opinion · 1/1

Student responses

1. NA tells you nothing and will not give a valid information about the correlation between the two; 0 is unreasonable because adding zero is kind of like biased number
2. They did not rate for oppenheimer. We can replace na, but not zero bc zero is a valid data in the dataset.
3. We know that they reviewed at least 1 of the 2 movies and the associated rating as well as we know what user rated them. What we don't know is what other ratings they have given to othr movies or the ratings of movies they havent seen. Replacing with 0 is not a good idea because it will skew our results
4. we know their barbie and oppenheimer ratings and whether they rated one or both movies. replacing the missing values with 0 would be bad because it would make it seem like because they didnt rate it they did not like it
5. We know what the rated Barbie, but not what they rated Oppenheimer. Replacing with zero is not acceptable since N/A says nothing concrete about their opinions.

Student responses

1. Replacing the missing values with zero would not be reasonable because it would look like people gave Oppenheimer 0 stars when that wouldn't necessarily reflect their thoughts on the movie. We don't know much about these people's ratings other than they chose not to rate Oppenheimer. They could've seen the movie and maybe liked it but we don't know that for sure.
2. These people only rated 1 of the films
3. we know about these people's rating for the Barbie, and we don't know about the ratings for the oppenheimer. it is not reasonable to replace it with zero, because it will affect when we compute such as means or sth
4. We know eir ID and Barbie rating, but no Oppenheimeri rating

Student responses

1. I want to drop those values which are NA, because these values wouldn't help us, so it is better to drop, because the we are comparing the rating from 1 to 5, anything else would create bad e
2. We know that these individuals had a rating for barbie but not Oppenheimer (possibly because they did not watch the movie). These values should not be replaced, instead, I think that these users should be dropped from the comparison.
3. we know they rated barbie but not oppenheimer. No, replacing with zero is unreasonable because it affects the relationship of correlation between them. inner_join may be better to remove rows with na

Does this plot indicate that people who like *Barbie* also tend to like *Oppenheimer*? Dislike? Explain your reasoning.

0 anonymous responses

No responses were submitted.

Replace each `____` and run your code. Submit your completed code and the percentage you found.

70 anonymous responses

Reported computed percentage Gives only the final numeric percentage result without showing the code or intermediate expressions.	17
Nonspecific or minimal attempts Very short or unclear responses offering non-code fragments, single words, or evidently incorrect function usage without a coherent pipeline.	12
Plain-text suggestions for blanks Lists suggested replacements in plain English, naming movies and indicating which blank corresponds to which replacement without showing code.	9
Concrete expressions for blanks Provides specific code expressions for the blanks, including conditional comparisons and a final ratio calculation.	7

Response themes for question 9

Malformed filter arguments — 7

Presents a pipeline with filter called using bare column names or missing arguments and an empty sum(), indicating confusion about proper filter/summarize syntax.

Single conditional expression — 7

Provides a lone conditional expression intended for a filter call, specifying a threshold comparison without the rest of the pipeline.

Code structure with multiple placeholders — 6

Writes the full pipeline structure including filter, summarize, and mutate but leaves several expressions unspecified (placeholder underscores) to be filled in later.

Response themes for question 9

Movie-specific pipeline with placeholders

5

Specifies filtering for a particular movie then includes summarize and mutate steps but leaves some expressions unspecified (placeholder underscores).

Student responses

1. 0.845 recommendation <- both |> filter(barbie_rating >= 4) |> summarize(people = n(), liked = sum(oppenheimer_rating >= 4)) |> mutate(percent_liked = liked / people) recommendation
2. 3.29
3. 39
4. 40%
5. 40%
6. 50%
7. 72
8. 84.5
9. 84.5

Student responses

1. 84.5
2. 84.5%
3. 84.5%
4. I had a percent_liked of 84.5
5. recommendation <- both |> filter(barbie_rating >= 4) |> summarize(people = n(), liked = sum(oppenheimer_rating >= 4)) |> mutate(percent_liked = liked / people) recommendation 84.5%
6. recommendation <- both |> filter(barbie_rating >= 4) |> summarize(people = n(), liked = sum(oppenheimer_rating >= 4)) |> mutate(percent_liked = liked / people) recommendation percent_liked = 0.845
7. recommendation <- both |> filter(barbie_rating >= 4) |> summarize(people = n(), liked = sum(oppenheimer_rating >= 4)) |> mutate(percent_liked = liked / people) recommendation, 84.5%
8. recommendation <- both |> filter(barbie_rating >= 4) |> summarize(people = n(), liked = sum(oppenheimer_rating >=4)) |> mutate(percent_liked = liked/recommendation) recommendation 84.5%

Student responses

1. /
2. a
3. answer for credit
4. Filter
5. filter(barbie_rating > 4) |> summarize(people = n(), liked = sum(oppenheimer_rating)) |> mutate(percent_liked =) 10%
6. filter(both$movies) n(10), sum(10) percent_liked(50)
7. I dont know
8. i got 213/1880 recommendation <- both |> filter(oppenheimer_rating >= 4 & barbie_rating >= 4) |> summarize(people = n(), liked = sum(oppenheimer_rating, barbie_rating)) |> mutate(percent_liked = liked / people)
9. Idk I didn't quite finish, but about 40 percent liked both

Student responses

1. Not sure
2. recommendation <- both |> filter(barbie_rating >= 4) |> summarize(people = n(), liked = sum(oppenheimer_rating >= 4)) |> mutate(percent_liked = (liked / people) * 100) recommendation
3. x

Student responses

1. I am still trying to figure it out
2. recommendation <- both |> filter(barbie >= 4) |> summarize(people = n(), liked = sum(oppenheimer >= 4)) |> mutate(percent_liked = liked/people) recommendation
3. recommendation <- both |> filter(barbie_rating > 4) |> summarize(people = n(), liked = sum(___)) |> mutate(percent_liked = ___) recommendation
4. recommendation <- both |> filter(barbie_rating >= 4) |> summarize(people = n(), liked = sum(oppenheimer_rating >= 4)) |> mutate(percent_liked = liked/people) recommendation The final percentage was 84.5
5. recommendation <- both |> filter(barbie_rating >= 4) |> summarize(people = n(), liked = sum(oppenheimer_ratings >= 4)) |> mutate(percent_liked = liked / people) recommendation
6. recommendation <- both |> filter(barbie_ratings == 4) |> summarize(people = n(), liked = sum(barbie_ratings == 4)) |> mutate(percent_liked = ___) recommendation
7. recommendation <- both |> filter(rating_counts) |> summarize(people = n(), liked = sum(oppenheimer_rating >= 4)) |> mutate(percent_liked = liked / people) recommendation. I found 84%

Student responses

1. the first blank is Barbie, Oppenheimer the second blank is something like rating the third blank is 4
2. title = "barbie"

Student responses

1. A
2. barbie_rating >= 4 oppenheimer_rating >= 4 100 * liked / people
3. barbie_rating >= 4 oppenheimer_rating >= 4 liked / people
4. In the sum I would put how many people rated higher than 4. percent liked will need to be liked over the total number of people to get the percentage
5. recommendation <- both |> filter(barbie_rating > 4) |> summarize(people = n(), liked = sum(oppenheimer_rating > 4)) |> mutate(percent_liked = liked / people * 100) recommendation
6. recommendation <- both |> filter(barbie_rating >= 4) |> summarize(people = n(), liked = sum(oppenheimer_rating >= 4)) |> mutate(percent_liked = 100 * liked / people) 84.5 percent
7. recommendation <- both |> filter(barbie_rating >= 4) |> summarize(people = n(), liked = sum(oppenheimer_rating >= 4)) |> mutate(percent_liked = liked / people) recommendation

Student responses

1. ecommendation <- both |> filter(rating_counts) |> summarize(people = n(), liked = sum(4)) |> mutate(percent_liked = ___)
2. recommendation <- both |> filter(barbie_r) |> summarize(people = n(), liked = sum(___)) |> mutate(percent_liked =)
3. recommendation <- both |> filter(barbie_rating > 4 & oppenheimer_rating > 4) |> summarize(people = n(), liked = sum(user_id)) |> mutate(percent_liked = size(both)) # recommendation # both
4. recommendation <- both |> filter(barbie_rating >= 4) |> summarize(people = n(), liked = sum()) |> mutate(percent_liked = sum)
5. recommendation <- both |> filter(oppenheimer_rating, barbie_rating) |> summarize(people = n(), liked = sum("barbie")) |> mutate(percent_liked = "barbie") recommendation
6. recommendation <- both |> filter(oppenheimer_rating, barbie_rating) |> summarize(people = n(), liked = sum()) |> mutate(percent_liked = liked/people) recommendation
7. recommendation <- both |> filter(user_id == oppenheimer) |> summarize(people = n(), liked = sum(4)) |> mutate(percent_liked = oppenheimer) recommendation

Student responses

1. barbie rating >= , oppenheimer rating > = 4,
2. barbie_rating >= 4
3. filter for barbie_rating
4. filter movie erating to be above 4 and then summarizze the sum of the people who liked that and then add that as a table
5. filter(barbie_rating >= 4) summarize(people = n(), liked = sum(oppenheimer_rating >= 4)) mutate(percent_liked = liked / people * 100)
6. put in oppenhiemer_rating
7. recommendation <- both |> filter(barbie_rating >= 4) |> summarize(people = n(), liked = sum(oppenheimer_rating >= 4)) |> mutate(percent_liked = liked / people) recommendation

Student responses

1. recommendation <- both |> filter(___) |> summarize(people = n(), liked = sum(___)) |> mutate(percent_liked = ___) recommendation
2. recommendation <- both |> filter(barbie >=4) |> summarize(people = n(), liked = sum(oppenheimer>=4)) |> mutate(percent_liked = liked/people *100) recommendation
3. recommendation <- both |> filter(barbie_rating > 4) |> summarize(people = n(), liked = sum(8)) |> mutate(percent_liked =) recommendation
4. recommendation <- both |> filter(barbie_rating > 4) |> summarize(people = n(), liked = sum(oppenheimer > 4)) |> mutate(percent_liked = liked/people * 100) recommendation
5. recommendation <- both |> filter(barbie_rating >= 4) |> summarize(people = n(), liked = sum(___)) |> mutate(percent_liked = ___) recommendation
6. recommendation <- both |> filter(barbie_rating >=4) |> summarize(people = n(), liked = sum(oppenheimer_rating >=4) |> mutate(percent_liked = liked / people * 100) recommendation

Student responses

1. filter(title = "barbie") |> summarize(people = n(), liked = sum(___)) |> mutate(percent_liked = ___)
2. recommendation <- both |> filter(barbie_rating >= 4) |> summarize(people = n(), liked = sum(oppenheimer_rating >= 4)) |> mutate(percent_liked = liked / people * 100) recommendation percentage = 84.5%
3. recommendation <- both |> filter(barbie_rating >= 4) |> summarize(people = n(), liked = sum(!) |> mutate(percent_liked = ___) recommendation
4. recommendation <- both |> filter(movie_id == "oppenheimer") |> summarize(people = n(), liked = sum(if rating>=4)) |> mutate(percent_liked = oppenheimer) recommendation
5. Start by keep only the rows in both where Barbie has at least 4 stars and Oppenheimer has at least 4 stars. Then count how many people liked Barbie, and among those, how many also liked Oppenheimer. Divide the second count by the first count and multiply by 100 to get the percentage. If you share your code template with the blanks, I can help you fill them in.

Would you recommend *Oppenheimer* to your roommate? Use the fraction and the heatmap to explain your answer. What uncertainty remains?

88 anonymous responses

Recommend due to high overall liking Respondents recommend the film based on the dataset showing a large majority liked it (approximately 85% or generally high ratings).	36
Explicit percent and heatmap explanation States the 84.5% fraction and mentions the heatmap correlation or table explicitly as justification for recommending.	25
Other responses Responses the model could not place reliably.	16
Other responses	8

Response themes for question 10

Concern about missing or NA ratings

3

Focuses on uncertainty arising from incomplete data or many missing ratings affecting confidence in the 84.5% estimate.

Student responses

1. i would reccomend. 84% of people who likec barbie also liked oppenheimer. there is still the 16% uncertainty
2. I would recommend Oppenheimer to my roommate because approximately 84.5% of Barbie fans who rated both movies gave Oppenheimer four stars are more, which is above 75%
3. I would recommend oppenheimer to my roommate, as 84% of people who liked barbie, aka gave it more than 4 starts, also liked oppenheimer, meaning they also gave it more than 4 stars. The uncertainty remains in where people who watched barbie did not also watch oppenheimer, so they never rated both movies.
4. I would recommend Oppenheimer to the roommate because the percentage of people that liked both movies was greater than 75%, as well as looking at the heatmap, there seemed to be a positive moderate correlation between the two movies and their ratings. That being said, we could always use a t-test to determine the true mean if needed.
5. I would, because the number was 84.5% which is greater than 75%.
6. Since 84.5% barbie fans who rated movies also liked oppenheimer I would recommend to my roommate

Student responses

1. The outcome is about 84.5% of users which means that Yeah most people would be recommending it and we should probably watch both now. I guess the uncertainty is whether does this really cooreelate and also does the NA people have an effect
2. yes
3. Yes
4. Yes because the percentage was greater than 75%
5. yes cuz its a good movie
6. Yes I would because greater than 80% of people who liked Barbie also liked Oppenheimer. There is still some uncertainty among specific individuals liking both movies but a majority like both.
7. Yes I would i twas rated highly
8. Yes I would reccomend it to my roomate because there were 84.5% of Barbie fans who gave Oppenheimer 4 stars or more
9. yes i would recommend oppenheimer to them

Student responses

1. Yes i would since 84% of people liked both,
2. Yes there are 84% in barbie also like oppen
3. yes, 84.5 rated oppenheimer highly
4. Yes, 84.5%. One uncertainty is this is self reported data
5. Yes, 84%. The heatmap also shows that people who like Barbie tend to like Oppenheimer
6. Yes, 85 is greater than 75, and the heat map shows the popularity of Oppenheimer amongst Barbie fans. Uncertainty only reminds for a bigger pool of data
7. Yes, because 84.5% is greater than 75% and that is what we found to be the data point
8. yes, because abt 85% people liked both
9. yes, because if at least 75% of Barbie fans like oppenheimer, it is large likely that the roomate will like oppenheimer since he/she is a Barbie fan

Student responses

1. Yes, because it has a higher star rate than barbie, and we got it from the people who watched them at the same time
2. Yes, I would since the fraction is over 75%.
3. Yes, it seems like roughly 84% of people who like Barbie also like Oppenheimer. The uncertainty exists in the sample size and user criteria for rating
4. Yes, many 4 and up
5. Yes, since 84.5 percent of people who liked Barbie also liked Oppenheimer, we should recommend it.
6. Yes, since the amount we observed was over 75%
7. Yes, since the fraction is 0.845, the heat map shows that the bulk of people enjoyed both movies.
8. Yes, the fraction of Barbie watchers who also liked Oppenheimer is over 75%. The heatmap also showed that those who liked Barbie tended to like Oppenheimer too. However, different viewers may have different opinions despite this relationship.
9. Yes, the mean value was greater than .75 so I would recommend Oppenheimer based on the data from this table

Student responses

1. Yes, there is a good chance they will like it
2. Yes. The rating of Oppenheimer is really high (almost 5), so I can surely recommend to my roommates. However, uncertainty arises that the raiting depends on the type of people responding to the rating.
3. yes. there is still around 15% uncertainty (people who liked barbie but not oppenheimer)

Student responses

1. # A tibble: 1 × 3 people liked percent_liked <int> <int> <dbl> 1 252 213 0.845 So maybe I should recommand it as there's a possibility of more than 80 percent of them liking it
2. 84.5% of people who liked barbie also liked oppenheimer so I would recommend it.
3. I would recommend it as 84.5% who liked Barbie also like Oppenheimer. Uncertainty remains in noise. This is a hard rule, and my roomate may be part of that 15.5% who would not like Oppenheimer given that they like Barbie.
4. I would recommend Oppenheimer to my roommate, because 84.5% of those in the survey who liked Barbie also liked Oppenheimer, which is greater than 75%. There may be some uncertainty depending on the sample size.
5. We should recommend Oppenehimer because more than 75% of fans, 84.5% of barbie fans gave both movies 4 star ratings of more. we are still unsure what people who did not rate barbie think about oppenheimer
6. We should recommend oppenheimer. The percent was above 75 percent and the heatmap was hot top-right. We do not know if they are necessarily correlated because we did not do statistical testing.

Student responses

1. Yes because 84.5% of the people who rated Barbie at 4 or more also rated Oppenheimer at 4 or more which meets the 75% threshold
2. Yes because the percentage is greater than 75. the undertaintlty is that
3. yes since our research told us almost 85% like both. Thea uncertainties are only the people who only watched one of the movies, since they havent watched both, we dont know if they like both or just the one they have watched
4. yes you would as 85% of people who watched babrbie enjoyed oppenheimer
5. Yes, 84.5 percent of Barbie fans rated Oppenheimer with 4 stars or more as well. It's uncertain if some of this correlation is just noise because some raters are just high raters on avg.
6. yes, because our value of people who liked barbie who also liked oppenheimer was almost 10 percent higher than the 75% threshold. The only uncertainty would be in the other 15 percent of people who did not like oppenheimer

Student responses

1. Yes, because the % of Barbie raters who ranked above a 4 and thought Oppenheimer was above a 4 was 84.5% which exceeds our 75% cap. However we did have a moderate correlation so some data remains variable
2. Yes, because the fraction is 84 > 75, and the heatmap shows a positive correlation between these two movie ratings
3. Yes, because the heat map shows that most people who liked Barbie also liked oppenheimer, although not everyone liked both, so there is still uncertainty that this will actually be true.
4. Yes, because the percentage os 84.5%; but there is still uncertainty about individual variance
5. Yes, I would recommened Oppenheimer since at least 75 percent of the Barbie fans who rated both movies gave it four start or more.
6. Yes, itss like almost 85 perrcent and heatmap shows storng corelation too
7. yes, more than 75% of barbie fans who rated both movies gave it four stars or more, we still dont know if your roommate will like it.
8. Yes, since a majoriyt of poeple who rate barbie high rate oppenhemier high as well

Student responses

1. Yes, since we concluded that 84.5 percent of Barbie fans who rated both movies gave four starts or more. The uncertainty probably lies in whether people who only watched oppenheimer gave it a higher rating, since we sampled oppenheimer ratings from barbie fans.
2. Yes, the heatmap shows us that at least 75% of the barbie fans who rated both movies gave it four stars or more
3. Yes! this is because the heatmap shows a good portion of barbie respondents also rating oppenheimer as a 4 star or more film. There is a relevant, positive correlation there
4. Yes! You would want to recommend Oppenheimer because it was about 85%. There does still exist about 15% uncertainty but I think it still makes sense to show it.
5. Yes. The percentage is above 75%, and the heatmap shows that many people who liked Barbie also liked Oppenheimer. There is still uncertainty because some users did not rate both movies and individual preferences may differ.

Student responses

1. I would not recommend Oppenheimer to my roomate becuse only ~61% of people who gave Barbie 4 stores or more also gave Oppenheimer 4 stars or more, and our threshold was
2. I would recommend it, since 84.5% who liked barbie also liked it. There still is uncertainty about
3. I would recommend Oppenheimer to my roommate because 84.5 percent of Barbie fans who rated both movies gave Oppenheimer four stars or more, which is within the at least 75 percent threshold. The uncertainty that remains is that my roommate may not actually like Oppenheimer.
4. I would. 84.5% liked it
5. Since 84.5% > 75%, we can recommend it. However, we also need to mention the potential outliers from our heatmap.
6. We should put on a different film because only 45% of people who rated Barbie 4 or more stars also rated oppenheimer four stars or more. The heat map still shows a positive correlation though.
7. Yes I would recommend Oppenheimer. About 84.5% of people who liked Barbie like Oppenheimer, however, there is uncertainty of other aspects like background and experience that might effect scores.

Student responses

1. Yes I'd reccomend because high percentage
2. Yes, because though the theme looks not very interesting, a lot of student who like Barbie, a very different type give it a high rate, which means it is meaningful.
3. Yes, fraction is > .75, so is recommended. Heatmap also shows high ratings of both at same time. Uncertainty that remains is reverse.
4. Yes, I would recommend Oppenheimer to my roommate because the fraction and the heatmap suggest that those that like Barbie also like Oppenheimer, so I would expect my roommate to have a decent chance at liking Oppenheimer.
5. Yes, I would recommend Oppenheimer. About 84.5% of the Barbie fans who rated both movies also liked Oppenheimer. But there is still uncertainty.
6. Yes, we found that 85% of barbie fans liked oppenhiemer (per our definition). However, there is still some noise (we did not test for signifigance). Also, i didn't like oppenhiemer so i wouldn't actually recomend
7. Yes. From the fraction, 84.5% of people who liked Barbie also liked Oppenheimer. The heatmap also shows a positive correlation between barbie ratings and oppenheimer ratings.

Student responses

1. Yes. The fraction gives 0.875, which is greater than 75%. The heatmap also shows most ratings concentrate at a high level. The uncertainty is we haven't seen data that rates oppenheimer but not barbie
2. Yoy, for sure, we should recommend people who like watch barbie with Oppnheimer, according from the heat map, the percentage of people vote for 4 are both high

Student responses

1. no
2. no, it is below 75%
3. Sure I think I would but to be sure we would have to see if this is just coincidental data and that maybe people who see big name movies just have low standards and like everything.
4. Yes we should as the heatmap shows that people like both movies around the same amount.
5. Yes, I would reccomend it because over 75% of Barbie fans also like Oppenhiemer. However, there is still a lot of scatter in the rankings, leading to a lot of uncertainty in our ability to predict our roommates response.
6. Yes, I would recommend Oppenheimer because the percentage of Barbie fans who also liked Oppenheimer is above the 75% cutoff, and the heatmap shows many ratings clustered high for both movies. There is still uncertainty because our roommate could have different preferences than the overall group.
7. Yes, it seems like there is a correlation between rating Barbie and rating Oppenheimer.
8. yes, looking at the previous calcuation finding how many percentage of people liked openheimler given that they liked barbie, the number is high enough to justify this.

Student responses

1. The number is 86%. The uncertainty is if the roommate hasn't rated barbie
2. Yes I would recommend it becuse 84.5% of people who rated both gave 4 stars. The uncertainty that remains is that less than half of the barbie watchers also rated oppenheimer so we need to figure out why those wer NA
3. Yes we should put it on because a huge number of people liked both oppenheimer and barbie. the uncertainty remains because maybe there is a movie we would like even more than oppenheimer

What table would let you estimate the average rating difference between two different films and run a `.test()`? Write a short plan for how you would construct it starting from `sequel_data` by working backwards.

88 anonymous responses

Pairwise ratings with differences — 43

Proposes constructing a table listing user IDs, each user's ratings for the two films, and the per-user rating difference to allow computing an average and running a test.

Aggregate means then t-test — 12

Suggests comparing average ratings of each film and running a t-test on group means without detailing per-user pairing.

Join two tables then pivot to pairs — 7

Explicitly proposes making separate rating tables for each film and joining them so each user occupies one row with both ratings.

Response themes for question 11

Match users to movies then pivot — 7

Describes joining or matching user IDs to their ratings for two specific sequels, arranging rows per user and columns per movie (or pivoting) to compute differences.

Unclear or missing plan — 6

Responses that do not provide a meaningful plan or specify a table structure.

Two-way frequency table — 5

Suggests using a contingency or two-way frequency table (counts) to compare films.

Binary liked/not-liked table — 4

Uses liked/not-liked indicators for each film (binary) to form a two-by-two table for comparison rather than numeric ratings.

Response themes for question 11

Filter for paired users then visualize

4

Starts by filtering to users who rated both original and sequel, align their ratings for paired comparison, and then create a plot.

Student responses

1. -filter out the two movies we want to compare -create a table for each movie iteration that includes the person id and the movie rating -join the tables by person id -subtract the original row from the sequel row
2. A table featuring only people who have watched both Shrek and Shrek 2 would allow you to get there. I'd work backwards by getting the table of this information and doing a summary.
3. A table that would let me estimate the average rating between two different films and run a t test would have the user_id in one column, the rating for the original in another column, and the rating for the sequel in a third column. I would filter by user id and create two new variables that only have the ratings for an original film and its sequel. I think I would need to filter the titles in order to create these new variables.
4. a table where one row is a user who has seen both movies, with columns for the user id, both movies' individual ratings, and the rating difference with positive or negative signage as its own column.

Student responses

1. a table with people who have seen both movies. We would run a hypthesis test with the null meaning the two means are the same and run a two-sided t test
2. a table with the ratings of both movies, join tables by user id and then subtract one rating from another, getting rid of nas
3. A table with user_id, the two ratings, and the difference between the ratings.
4. A table woth the rating of the first move and a rating of rhte seocnd move and then the dfifferenc ebetween the 2nd and first, THis will allow us to see if people like the sequel better or not
5. Calculate difference in a new column, then take the average difference among all users in a pair of movie and sequel
6. Construct a table with both films. having two columns with their ratings. Ans subtract each other, stored the new information in a new column, after storation draw a scatter plot
7. create a joined data table of people who saw both. understand the variance of each groups then run the t test

Student responses

1. Create a new table contains the only individuals who rated both the original and the sequel. Run a t test on this data to see if it is statistically significant.
2. filter original table, Each row contains one person’s two ratings. use t test with the two columns
3. Filter sequel data to get two datasets with each of the two movies you are looking at. Mutate columns with original rating, then sequel rating, Join both of them by user_id. Summarize this new table by taking the mean of both movies and run a t.test to see if they are different.
4. Get a table with a user's ratings for both the original movie and the sequel. To construct this first I would make a table with all users data for the original and then a table with all ratings for the sequel and join them together on user_id then I can find the average difference and run a test.

Student responses

1. get the rating for original and the ratings for sequel and put them into the same table matched by user_id, add a column for the rating difference and then calculate the average, if the average is positive then people tends to like the sequel. difference = sequel - original
2. I would create a table with one row per user and separate columns for the original movie rating and the sequel rating. I would filter sequel_data for the two movies, separate their ratings, and join the two tables by user_id so that each person's ratings are matched. Then I would calculate the difference as sequel_rating - original_rating and use these differences to find the average and run the .test().
3. I would have to gather individual users' ratings of each film and attribute them to the correct user ID. I would then have to calculate the difference in ratings between the first movie and the sequel for each person. I would then have to compare these differences across each set of films (first + sequel) by averaging the differences for each film.

Student responses

1. i would make a table where each row would contain one's person's two ratings.
2. I would make a table with one row per person who rated both movies, with one column for the original rating and one for the sequel rating. I would filter sequel_data to the two movies, make a separate table for each movie, and then join them using user_id. Then I could calculate sequel rating − original rating for each person and use those paired ratings in the .test().
3. I would need to match the movie title to the ratings first for the movies and sequels. Then I would need to match the originals to the sequels in a table with the respective ratings for each film. After I need to filter out only the people who have seen both films, then I can do my analysis
4. I would separate the data in the table to ratings for the first movie and then ratings for the second movie by user.id. Then, I would eliminate users who have only rated one of the movies. I would join these data together by user.id and find the percentage of users that ranked the second movie above the first movie.

Student responses

1. I would start from sequel_data and pair all the differences between ratings.
2. I would want to do a t.test for the difference in ratings, so I would want a table of user_id, shrek 1 rating, shrek 2 rating. I would filter sequel data to just shreks, then split into shrek 1 and 2, then join on user_id.
3. I would want to use a table that contains the ratings for both the original as well as the sequel, so that I could find the average rating, and then use a t.test() on the averages to determine a statistical conclusion.
4. Load the Data. Stick to the ratings by each person. Make a table. Match the people who rated both. Calculate the percentage for both. Run the t-test and compare.
5. mutate() and create a new column/variable that is the difference in rating for the movies in a singular franchise that have sequels.
6. pair ratings with differences
7. pairwise ratings with differences

Student responses

1. Plan: split strings by regex, extract year, see if we can find some relation between names (common substring that isn't in an obvious corpus like the?), then get pairs of movies (sequels and originals) and their ratings, then run a t-test comparing the two groups.
2. Scatter plot would let us see how people voted normally. I would left group by movie id and then left group again by user id to get them all on the same table and then add the total ratings of the two movies and divide the number of people who liked the sequel more than the original by the total to get the percentage.
3. similar to before with barbie and oppenheimer, create variables and join them together and then compare the 2 ratings while in the same row with the user id
4. start with sequel_data and work backwards by building the data from the main movie dataset. check to see if there is a difference between the first movie and the second movie then you can see if people prefer the sequel over the first movie and then perform a join before doing that to show the comparison

Student responses

1. Table containing ratings of 2 different fils, filter by user id, rating, movie id,
2. take the ratings of a movie and make a table. Then take the ratings of the movie's sequel, and make a table. Then join those tables together using user_ID. Repeat process for all the pairs of movies. Then take a t.test comparing the average between the two movies ratings
3. the same scatterplot finding the average number for the ratings of both movies.
4. The table of rating from the same people to teo different films
5. The table would calc the diff betwween 2 movies in a series
6. The table would include the abs difference between the two chosen movie instead of listing out each movies rating. From the t test, we can observe a table with a pairwise rating with differences.
7. To estimate the rating between two different films, you would need to create a table using user_id as the connection, where the rating for each movie and its sequel is compared. This would be done by associating the correct movie_ids.

Student responses

1. We would need a table that has ratings for the movies, original and sequel. First we start with sequel_data and select the important rows, specifically the movie_id. Then we can assign movie _id to each movie in the ratings table and we can join these tables.
2. Would use the movie ids and then match those with scores, and subtract them, not sure what table it would be tho
3. You would need to join the ratings of the two movies based on the user ID, filter out NA responses, then take the average rating of the movies, and run a t-test based on your results.

Student responses

1. A table with number of average ratings of each film. I would need to first select the two films, and group by films, and then calculate the average number of each group
2. filter the data into two tables, one for each film, and then find the average rating for each film from the tables and compare the two using a .test.
3. Finding the mean rating for each film, and then joining the tables to see a side by side comparison
4. I would create a table for each film. Then I create tables that join each film and their sequel. Next I would make sure each table has na values for users that only rated one of the films so each table contains user_ids and their ratings for the first and second movies. Then I would use summarize to calculate the means of the sequel and first movies. Lastly, I would calculate the difference between the two means.
5. I would filter by each movie and tally up the ratings for each, then calculate the avg rating for each film, in a table with compare two movies and get the difference of their avgs

Student responses

1. I would filter the data to show only the two selected titles, then create separate variables with the mutate command for the rating of each title. I will then take the mean of both total ratings and run a t-test to compare the two.
2. i would find the average of ratings for one film and the average of ratings for the other film and run a t.test for a difference in means. i would use a left_join to lookup user_ids in one dataframe and get the other value into my table so they are combined. I would drop NAs and then run the avergaes. use summarize(mean)
3. I would join the tables using movie_id, then take the mean of the sequel and minus the mean of the original for a mean difference in stars column.
4. Look at sequel data and compute the averages for each original movie compared to its sequel, which can be used through a t test to see if the difference is statistically significant.
5. maybe separate the data based on sequel pairing in order to avoid having too many NA drops. then do exactly what we did before but three times for each of the sequel pairs.

Student responses

1. run a one higher sided t test comparing the mean score of the orginal and the sequal using the movie id to assign mean score. this would be form a joined list connected by uer id of people who have seen both movies
2. The table we need is a table of all users who have rated any of these films and compare there ratings

Student responses

1. i would do a joint table again, pulling shrek 1 and shrek 2 in a similar manner to what we did for barbie and oppenheimer. then i would run a test on the average difference in scores between the 2.
2. I would first find movie rating for into spider verse and across spider verse seperately and have the rating and the user id. than i would join by user id to compare the ratings of the two movies together by user id.
3. I would go movie pairing by movie pairing. I would join user_ids with movie ratings for each of the two movie sin the duo of movies. I woukd then scatter these two variables/movies. Original on bottom and sequel on y axis. and the analyze from there
4. I would use a scatter plot to see how they line up. Join by the user_id and check the individual differences
5. make a ratings table with the title and user id. Filter by title into new tables to make original and sequal ratings columns, then use an inner_join to combine and remove NAs so that every user response has ratings for both, then run a t.test on the means of these columns

Student responses

1. To do this, you would first have to join all of the sequels together. Then, uou could apply what we did between barbie and oppenheimer to run a t test
2. two tables for each movie with the rating and user. then join the two tables so each user is one row with both of their ratings for each movie.

Student responses

1. a scatter plot could be helpful for seeing the relationship between the films
2. connect the user_id with each movie, seperate the movies 2 by 2 with their sequels, match their rating to each movie
3. into <- named_ratings |> filter(title == "Spider-Man: Into the Spider-Verse (2018)") |> select(user_id, into_rating = rating) across <- named_ratings |> filter(title == "Spider-Man: Across the Spider-Verse (2023)") |> select(user_id, across_rating = rating) paired <- into |> left_join(across, by = "user_id") both <- paired |> filter(!is.na(into_rating) & !is.na(across_rating)) recommendation <- both |> filter(into_rating >= 4) |> summarize(people = n(), liked = sum(across_rat
4. Pull the movie rating groups from the data base. Join them with response id so each rating is paired.
5. rows are the different movie sequels and the col is the difference between the two ratings

Student responses

1. shrek <- sequel_data |> filter(title == "Shrek (2001)") |> select(user_id, shrek_rating = rating) shrek2 <- sequel_data |> filter(title == "Shrek2 (2004)") |> select(user_id, shrek2_rating = rating) paired <- shrek |> left_join(shrek2, by = "user_id") both <- paired |> filter(!is.na(shrek2_rating)) These are what I've done for now. Next I am going to count average rating of both films and conduct a t-test for statistical significance.
2. Use sum or mean to get the seuel's score and minus with original

Student responses

1. Filter for either movie, make seperste tables, joij on user id, find diff between user
2. how are you?
3. I would load in the datasets, then I would filter the tables and ensure that ratings are existent. Then i would join them by their IDs and compare
4. I would need to get a sampling distribution of both movies
5. I would run a test and work backwards to fidn the average rating difference.
6. x

Student responses

1. 1. load data 2. get the specific series1 and then series2 3. tidy them in two separated columns 4. calculates the percentages
2. a table of 2*2
3. a two way frequency table
4. If u use the preivous table saorting by user id and then run l.test on it
5. two way frequency table. i would, we can build it to compare individual user ratings between films. Using this rating, we can do a t-test

Student responses

1. a table of liked (4 or over)/not liked (under 4) for both films
2. get separate tables for the original and sequel for the two movies. do an inner join on user id, because we only care about people who have seen both. Get an average rating difference into a new col.
3. I mean the table would be the percentage, and youd have a table based by user id that joins the original rating with the sequel rating and you start by building two tables split by user id one for original and one for sequel then left join the sequel on then you can count and do your calculations
4. t.test(rating ~ film, data = film_ratings)

Student responses

1. A scatter plot would help you see the relationship between two films clearly. You would need to compile user Id and the ratings for films all into one row for each person.
2. I will start by filtering the data of the sequel and the original, compute them into the same model with the same user providing ratings for both movies. After that, I will generate a graph based on the data.
3. Lastly personal question to the professor why does he not like top gun maverick? What you would do is the make a table for just a particular beginning film(first film and its rating) then you would make another table that is for the rating of the sequel and then you would join both of those tables. From there you can mutate a new column which could be to calculate the difference or average. Then we can do this for each one.

Student responses

1. table with one row per person who rated both films and two rating columns, one for the original and one for the sequel. From sequel_data, you would keep the reviewer ID and movie title and stars, pivot so each film becomes its own column, and keep only rows with both ratings present. Then compute a new column for difference = sequel minus original, and use that column to estimate the average difference and run the test.

Submit the film pair your code, and the resulting table. Which film had the higher mean rating? Can you use `t.test()` to construct a a confidence interval for the difference? What's the p-value?

0 anonymous responses

No responses were submitted.