

AI, skills, and your career

Week 5, Lecture 10

<https://live.ds306.org>

October 1, 2026

Warmup: Halloween weather

```
a2weather |>  
  select(date, year, month, day, tmin, tmax, prcp) |>  
  head(3)
```

Predict the max temperature for Ann Arbor on **October 31, 2026** using a2weather. Use any method you like. Submit your forecast (with units) and explain your approach.

From Tuesday

- We saw declining employment among young workers.
- We considered several possible explanations.
- At the end, we introduced linear regression.

Today: what might AI change about the work you do, and the skills you need?

Four questions

1. Which jobs are most exposed to AI?
2. Are highly AI-exposed fields predicted to shrink?
3. Which skills are most associated with higher pay?
4. How much is critical thinking worth?

Q1: Which jobs are most exposed to AI?

AI exposure describes how much of an occupation's work AI could help perform.

Name an occupation you expect to be highly exposed to AI. Name an occupation you expect to be highly resistant to AI.

What does “exposure” measure?

- In a 2023 study, researchers assessed whether AI could substantially reduce the time needed for a task.
- Different professions were assigned a score of 0-100 based on how much of their work could be assisted by AI.

Scoring definitions and data.

AI exposure

exposure gives a score for each occupation, identified by `occupation_code`.

```
exposure |> head(4)
```

`exposure_percent` is the exposure score on a 0–100 scale.

Projected employment growth

bls gives the occupation's name and projected percentage change in employment from 2025 to 2035.

```
bls |>
  select(occupation_code, occupation, projected_growth) |>
  head(4)
```

We can match occupation codes to get the names. family groups computer/math, office/administrative, and other occupations for Q2.

Find the most exposed occupations

What are the five most AI-exposed occupations?

```
bls
```

```
# Combine the sources, then find the highest exposure score
```

- Join the tables by occupation code, keeping occupations found in both sources. Save the result as ai.
- Find the five most AI-exposed occupations in the matched table.
- Display their names

Which occupations have the highest scores?

```
ai <- inner_join(bls, exposure, by = "occupation_code")

ai |>
  arrange(desc(exposure_percent)) |>
  select(occupation, exposure_percent) |>
  head(5)
```

What kinds of tasks do these occupations have in common?

What about data scientists?

Let's look up their exposure score.

```
ai |>  
  filter(occupation == "Data scientists") |>  
  select(occupation, exposure_percent)
```

Where do data scientists rank?

- How many occupations have higher AI exposure?
- We could sort all 749 occupations by exposure and count down to data scientists.
- A **rank** records an occupation's position in a sorted list.

`min_rank()` adds a rank for each value. Use `desc()` to give the highest score rank 1.

```
ai |>
  mutate(exposure_rank = min_rank(desc(exposure_percent))) |>
  arrange(exposure_rank) |>
  select(occupation, exposure_rank) |>
  head(3)
```

Equal scores share a rank, as in 1, 2, 2, 4.

Find our place in the rankings

Where do data scientists rank among the 749 occupations for AI exposure?

```
ai
```

```
# Find the exposure rank for data scientists.
```

- Rank all occupations in ai by exposure. Give the highest value rank 1, with equal ranks for ties.
- Keep the row for Data scientists.
- Display the occupation name, exposure score, and rank.

Where data science stands

```
ai |>
```

```
  mutate(exposure_rank = min_rank(desc(exposure_percent))) |>
```

```
  filter(occupation == "Data scientists") |>  
  select(occupation, exposure_percent, exposure_rank)
```

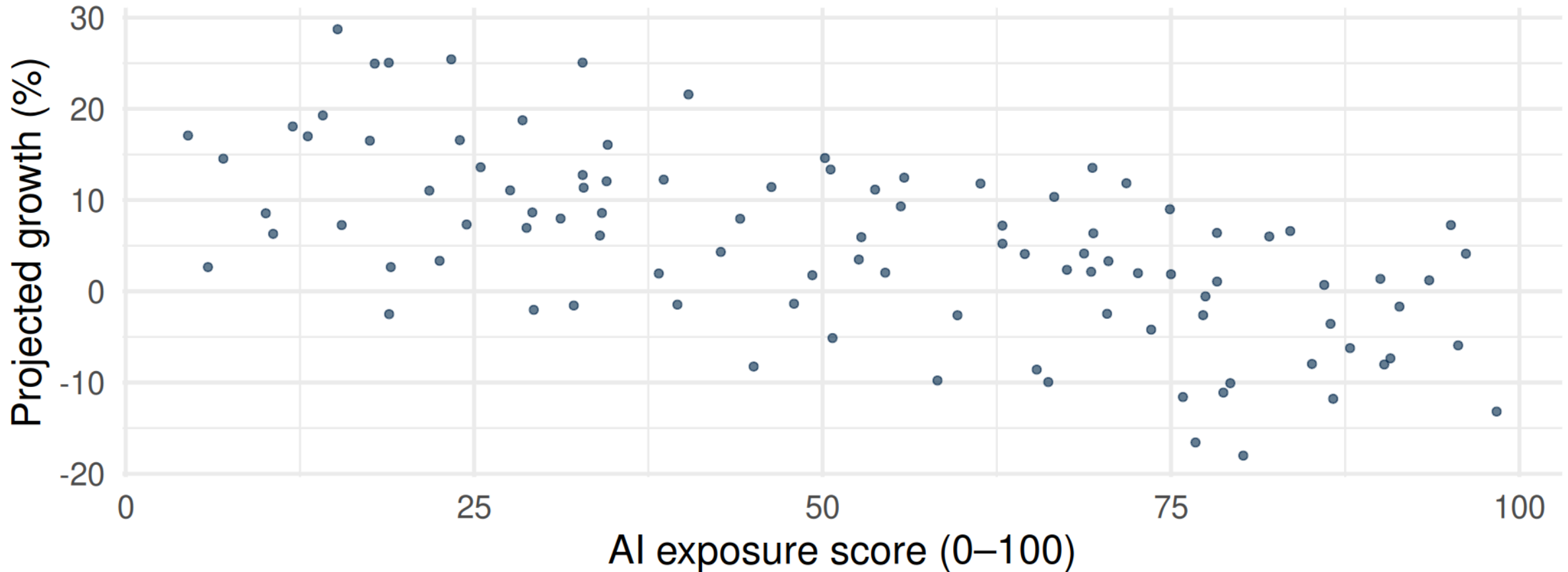
Q2: Are highly exposed fields predicted to shrink?

Data science has an exposure score of 75. What might that mean for employment?

Would you expect highly exposed occupations to grow faster or slower? Explain how AI could lead to the outcome you predict.

A possible pattern

If greater AI exposure goes with slower growth, we might see this:



Are AI-exposed jobs projected to grow?

```
ai
```

```
# Plot projected growth against AI exposure.
```

Use `ai` to make a scatterplot of projected growth against AI exposure, with one point per occupation. Label the axes.

Run your code, then submit your plot. We will look at some of the plots together.

Investigate highly exposed occupations

```
ai
```

```
# Which kinds of highly exposed work are projected to grow
```

- Keep occupations with exposure scores of at least 75.
- For each family, make a boxplot, and add dots for the individual occupations.

Which highly exposed careers are projected to grow?

```
ai |>

  filter(exposure_percent >= 75) |>

  ggplot(aes(projected_growth, family)) +

  geom_vline(xintercept = 0, color = "gray60") +
  geom_boxplot(outlier.shape = NA, width = 0.4) +

  geom_jitter(aes(color = family), width = 0, height = 0.08, size = 2) +

  labs(x = "Projected employment growth, 2025-35 (%)", y = NULL) +
  theme(legend.position = "none", axis.text.y = element_text(size = 12))
```

What would you tell a data science major?

What would you tell a data science major who thinks high AI exposure means their field will shrink? Use the plot to support your answer.

Q3: Which skills are most associated with higher pay?

If you could spend a year getting much better at one skill, which would you choose? Why do you think employers would value it?

How could we investigate that?

- We'll use **pay** as one measure of how the labor market values work.
- For each occupation, we have its median annual pay and the skills it requires.
- Which skills are most strongly associated with higher-paying occupations?

Two sources of data

- The **2024 American Community Survey** records people's occupations and earnings.
- We calculated median annual pay for full-time, year-round employees aged 25–64 in each occupation.
- ****O*NET**** rates how important each skill is to each occupation, from 1 to 5.
- Our tables cover **489 occupations and 35 skills**.

What skills can we investigate?

```
skills |> group_by(skill) |> summarize(occupations = n()) |> head(8)
```

There are also ratings for programming, mathematics, speaking, negotiation, repairing equipment, and many others.

Our two tables

```
skills |> select(occupation_code, skill, importance) |> head(3)
wages |> select(occupation_code, pay) |> head(3)
```

- `skills` has one row per occupation and skill.
- `wages` has one row per occupation. `pay` is in thousands of dollars.

Investigate a skill

Choose a skill. How would you investigate whether occupations requiring more of it tend to pay more? Describe a table or plot you would make.

Investigate your chosen skill

```
skills |> inner_join(wages, by = "occupation_code")  
  
# Join the tables, choose a skill, and plot.
```

Choose one skill and use `skills` and `wages` to make a scatterplot of its importance against annual pay, with one point per occupation. Label both axes and submit your plot.

Let's look at critical thinking

```
skill_pay <- inner_join(skills, wages, by = "occupation_code")
thinking <- skill_pay |> filter(skill == "Critical Thinking")

thinking |>
  ggplot(aes(importance, pay)) +
  geom_point(alpha = .4) +

  labs(x = "Critical-thinking importance (1-5)", y = "Annual pay ($1,000s)")
```

Improve the plot

The highest-paid occupations stretch the y-axis and compress most of the points. Can you make the pattern among the other occupations easier to see while keeping the high-paid occupations visible? Edit the plot and submit it for the gallery.

```
thinking |>
  ggplot(aes(importance, pay)) +
  geom_point(alpha = .4) +
  labs(x = "Critical-thinking importance (1-5)", y = "Anr
```

Improve the vertical scale so the relationship among most occupations is easier to see. Keep all occupations visible, then submit your plot.

How can we look across all 35 skills?

- We could make 35 scatterplots. It would help to summarize each relationship with a number.
- **Correlation** measures the direction and strength of a linear relationship, from -1 to 1 .
- `cor(importance, pay)` calculates it for our two columns.
- Larger positive correlations indicate skills more closely associated with higher pay.

```
thinking |> summarize(correlation = cor(importance, pay))
```

Which skills rise to the top?

What are the top 3 skills that are most positively associated with higher pay?
Negatively?

```
skill_pay
```

```
# Calculate a correlation for each skill, then rank the r
```

- Group the rows by skill.
- Calculate the correlation between importance and pay within each skill.
- Sort the correlations from highest to lowest and display the first

The strongest associations

```
skill_ranking <- skill_pay |>
  group_by(skill) |>
  summarize(correlation = cor(importance, pay))

skill_ranking |> arrange(desc(correlation)) |> head(8)
```


Looking back

Find the skill you chose at the beginning. Where does it appear in the ranking? Is this what you expected?

Q4: How much is critical thinking worth?

We'll use regression to estimate how annual pay varies with the importance of critical thinking across occupations.

How much more do these jobs pay?

For critical thinking, fit a **regression line**:

`pay = intercept + slope * importance`

- The **slope** gives the expected pay difference for a one-point difference in importance.
- The intercept gives the line's predicted pay at zero.

Regress pay on importance

Use thinking. Save your model as `pay_model`.

```
thinking  
  
# Fit the model, inspect its coefficients, and draw the f
```

- Fit pay using importance as the predictor, and save the model.
- Display the intercept and slope.
- Plot the observations and add the fitted regression line.

Interpret the slope

```
pay_model <- lm(pay ~ importance, data = thinking)
coef(pay_model)
```

Two occupations have critical-thinking importance scores of 3 and 4.
What difference in annual pay does this line predict, in dollars?

How much variation does the line explain?

- **R-squared** measures the fraction of pay variation accounted for by the fitted line.
- `summary( )` collects model information; `$r.squared` selects this value.

```
summary(pay_model)$r.squared
```

One occupation stands out

```
thinking |>  
  arrange(desc(pay)) |>  
  select(occupation, importance, pay) |>  
  head(3)
```

If we refit the regression without the highest-paid occupation, what do you expect to happen to the slope and R-squared? Explain your prediction.

Refit without the highest-paid occupation

Keep `thinking` and `pay_model` for the original fit. Use new names for this fit.

```
thinking_without_top <- thinking |> filter(_____)
pay_without_top <- lm(pay ~ importance, data = _____)
coef(pay_without_top)
summary(pay_without_top)$r.squared
```

- Save all but the highest-paid occupation as `thinking_without_top`.
- Regress `pay` on `importance` using those occupations; save the model as `pay_without_top`.
- Display the new model's coefficients and R-squared.

What changed?

```
thinking_without_top <- thinking |> filter(pay < max(pay))  
  
pay_without_top <- lm(pay ~ importance, data = thinking_without_top)  
  
coef(pay_without_top)  
summary(pay_without_top)$r.squared
```

- The slope falls from about \$51,000 to \$47,000 per importance point.
- R-squared rises from 39% to 49%.

What if we regressed on problem solving instead?

What do you expect the slope to be if we regress pay on **Complex Problem Solving** importance instead of **Critical Thinking** importance? Why?

Fit the problem-solving model

```
problem_solving <- skill_pay |> filter(skill == "Complex Problem Solving")  
problem_model <- lm(pay ~ importance, data = problem_solving)  
coef(problem_model)
```

The slope is about \$46,000 per importance point, close to the critical-thinking slope of about \$51,000. Across occupations, those two skill ratings are strongly correlated ($r = .90$), so each one captures some of the same pay pattern.

Would using both skills improve the prediction?

Either skill alone accounts for about 39% of the variation in pay.

If we use both critical thinking and problem solving to predict pay, do you expect us to explain substantially more variation? Why?

We need two skill columns

Our regression needs one row per occupation, with pay and both skill scores.

```
both_skills |>
  select(occupation, pay, thinking, problem_solving) |>
  head(3)
```

In `skill_pay`, the two scores occupy separate rows. `pivot_wider()` can put them into separate columns. We'll use shorter names for those columns.

A regression with two predictors

In an `lm()` formula, `+` includes another predictor:

```
both_model <- lm(pay ~ thinking + problem_solving, data = both_skills)
```

- The `thinking` slope gives the predicted pay difference for one more point of critical thinking at the same problem-solving score.
- The `problem_solving` slope gives the predicted pay difference for one more point of problem solving at the same critical-thinking score.

Fit both skills together

The supplied fold creates `two_skills` with the shorter skill names. Fill the next folds in order and use **Run code to current fold**.

```
two_skills <- skill_pay |>
  filter(skill == "Critical Thinking" |
         skill == "Complex Problem Solving") |>
  select(occupation_code, occupation, pay, skill, importance) |>
  mutate(skill = ifelse(
    skill == "Critical Thinking", "thinking", "problem_solving"
  ))

both_skills <- two_skills |>
  pivot_wider(names_from = occupation_code, values_from = pay)

both_model <- lm(pay ~ thinking + problem_solving, data = both_skills)

coef(both_model)
summary(both_model)$r.squared
```

- Reshape `two_skills` into `both_skills`, with one occupation per row and separate thinking and problem_solving columns containing importance scores.
- Fit pay using both skill scores, and

How much did the second skill add?

```
both_skills <- two_skills |>
  pivot_wider(names_from = skill, values_from = importance)
both_model <- lm(pay ~ thinking + problem_solving, data = both_skills)

coef(both_model)
summary(both_model)$r.squared
```

Both slopes fell substantially, while R-squared rose from 39% to 41%. How would you explain that? What does this suggest about interpreting the original \$51,000 slope as the value of critical thinking?

What does the regression line miss?

Return to our original critical-thinking model, `pay_model`.

`residual = observed pay - predicted pay`

- `predict(pay_model)` gives predicted pay for each occupation.
- Positive residuals identify occupations paying more than predicted.
- Negative residuals identify occupations paying less than predicted.

```
thinking <- thinking |> mutate(residual = pay - predict(pay_model))
```

Which occupations surprised us?

```
thinking
```

```
# Find occupations above and below the model's prediction
```

- Find the three occupations with the largest positive residuals.
- Find the three with the most negative residuals.
- For each group, display occupation, importance, pay, and residual.

Takeaways

- Data science is among the occupations most exposed to AI.
- It is also expected to experience the most employment growth over the next decade.
- High AI exposure can coexist with projected employment growth.
- Judgment, critical thinking, and problem solving are the most highly correlated skills with higher pay.
- (And are the skills we're trying to build in courses like DATASCI 306!)

Sources and data

- Ann Arbor weather records from Week 2.
- Eloundou et al., exposure study and published scores.
- BLS occupation projections.
- ACS 2024 microdata and O*NET.
- Skill ratings, occupation wages, and preparation notes.

Week 5, Lecture 10: AI, skills, and your career

Closed October 1, 2026 at 4:13 PM

23 class questions · 1183 anonymous responses · 0 student questions

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

Predict the max temperature for Ann Arbor on ****October 31, 2026**** using ``a2weather``. Use any method you like. Submit your forecast (with units) and explain your approach.

68 anonymous responses

Climatological average with units and rationale

28

Explains using historical October 31 maximum temperatures averaged from the dataset to produce a forecast; includes temperature in °C and equivalent °F.

Bare numeric prediction without method

13

Provides only a numeric temperature value with minimal context or explanation; no method described.

Bare Fahrenheit prediction without method

5

Provides a single temperature in degrees Fahrenheit with minimal context or explanation.

Implausible high-value guess without justification

5

Presents an unusually high temperature prediction that is inconsistent with October norms and provides no supporting method or data.

Response themes for question 1

Mixed/unclear multi-variable output**5**

Gives multiple weather variables and values (min, max, precipitation, snow) without explanation of method or units consistency.

Method stated without numeric forecast**4**

Describes the forecasting approach (using historical averages) but does not provide a specific numeric temperature prediction.

Range or min/max pair provided**4**

Gives both a low and a high (or implies a range) in Fahrenheit rather than a single max prediction, without a detailed method.

Regression-based numeric forecast with method**4**

Presents a numeric prediction derived from a year-over-year regression using prior-year max temperature and specifies the computed adjustment and resulting value in °C.

Student responses

1. 13 degrees C, I took three random Halloweens and averaged their Tmax
2. 13 degrees celcius since last year it was around there
3. 13.4
4. 13.4
5. 13.4 celsius; i averaged the highest temp recorded on halloween for all the years of data we have
6. 13.4 degrees
7. 13.4 degrees C when doing an average across all years of the 135 years of the data
8. 13.4 degrees celsius, I filtered using month as October and Day as 31 and then summarized the average temeprature. I also removed any N/A data inputs.
9. 13.7 by filtering all 10.31 after year 2020 and take the mean

Student responses

1. 13.7 degrees Celsius, found by averaging the mean tmax across oct 30 to oct 31 from 2000 onwards.
2. 14.32 degree Celsius, used by averaging max temperatures from the past 5 Halloweens
3. 15, from looking at previous tmax
4. 57°F, based on the historical average/median Halloween high temperature in Ann Arbor.
5. `a2weather |> select(date, year, month, day, tmin, tmax, prcp) |> filter(month == 10, day == 31) |> summarize(m = mean(tmax, na.rm = T))` 13.4
6. `a2weather |> select(date, year, month, day, tmin, tmax, prcp) |> filter(month == 10, day == 31)|> filter(tmax != 'NA')|> summarize(m = mean(tmax))`
7. Forecast: 55°F (about 13°C). Approach: filter a2weather to October 31 across all available years and take the median of tmax (median is robust to unusually warm or cold Halloweens). That historical median is right around 55°F, which also matches Ann Arbor's late-October climatology, so I use it as my point forecast.
8. Get the mean max for October 31st through the past years

Student responses

1. I averaged the last five years by their max temperature, and arrived at a prediction of 14.32 degrees celsius. I did not want to average the entire history, as global warming has led to general growth in max temperatures.
2. I calculated the average max temperature in Celsius from the week of Halloween last year, and got 13.5 in Celsius.
3. I predict that the max temperature in Ann Arbor on Halloween will be 14.6 degrees C. I did this by taking the average of the max temperature for each day over the two-week period before and after Halloween, and I did this for each year since 2000.
4. I think it will be around 14 degrees Celsius based on looking at values close to before and after halloween
5. I used the average maximum temperature (tmax) from previous October 31 observations in the a2weather dataset as my prediction for October 31, 2026. 13.4 degree C
6. I would predict about 13°C (55°F) by averaging the maximum temperatures from past October 31 dates in the dataset and using that as my forecast.

Student responses

1. It should be around 13.4 degrees celcius at the hottest point in the day. I used filter and summarize to find the mean on the specific date.
2. It should have a tmax of 13.4 degrees celsius
3. predict a maximum temperature of 14 celsius o I averaged the October 31 maximum temperatures from the most recent 30 available years in a2weather.
4. roughly 13 degrees celsius, but I'm still working on it! The approach I used was taking the mean from all previous years of october 31st.
5. use the mu of the pdf of the weather distirbutions

Student responses

1. 13.4
2. 13.4
3. 15 degree Celsius. I filter out the maximal temperature of Oct 31 from 2000 from the table and take the mean of these values.
4. 20
5. 20 C
6. 20 deg C
7. 24 degrees celcius. The data
8. I predict that the max temperature will be 22 degrees celcisu by averageing the 10 highest temps
9. I predict the max is 14 degrees celcius, as the average is 13.4 overall but the weather has likely gotten warmer due to global warming.

Student responses

1. I predict the max temperature to be 13.4 degrees Celsius. This approach filters by month == 10 and day == 31, then calculates the mean of max temperature from the data, ignoring missing values.
2. I used summarize and found the max was 13.4
3. The max temperature will be 56 degrees using average over all years
4. tmax in oct 2026 31 might be 20 celsius

Student responses

1. 60 degrees
2. 60 degrees fahrenheit since it's in the 60s for the first couple of days and it usually gets cooler by the end.
3. I predict that the high temperature will be around 55F, with a low of 41F, as those seem to be around the middle of recent years' temperatures.
4. I think it will be a low of 44 degrees (fahrenheit), with a high of 59
5. The max temperature will be 60 degrees Fahrenheit, which is about the temperature around that time of year.

Student responses

1. 85 degrees f, as we can increase head (30) or average their temperature.
2. I predict a max temp of 88 degrees.
3. I predict a maximum temperature of 23 °C. I used the average maximum temperature on October 31 from 2016 to 2025 to estimate this year's temperature.
4. i predict the max temp is 73 degrees fahrenheit. i would use a2weather by filtering by year and month and finding the max temp
5. If I had to guess, the max in October is 81 degrees.

Student responses

1. 56.1 degrees fahrenheit, average temperature over time
2. Find the mean tmax over all halloweens in the past 40yrs cause global warming
3. I would start looking at the averages of of the tmeperature plu or minus a week from october 31st and use that as a prediction to see the max temp
4. my prediction is 13.4 degrees celsius. i filtered the data to only include halloween of each year and took the mean temperasture of those days
5. tmin 0 tmax 19 prcp 1 inch of snow

Student responses

1. .13.3 C
2. filter down to all halloweens in past and average $t_{\max} - t_{\min}/2$
3. I would take the last years and see if there is any kind of linear or other trend I can find and use that to best predict the year 2026.
4. Use average temperature on October 31 from previous years.

Student responses

1. 20
2. around 50 - 70, typical range for the last 30 years
3. I took the data from 1970 on and averaged the Halloween dates and returned a tmax value of 13.8
4. The

Student responses

1. Going purely on the data of the prior year and a regression of the average day temperature year over year, I predict it will be 0.2 C hotter than the previous max temp, thus 11.3 degrees C
2. I guess 10°C , because I remember exactly when I when to the game on 11.1 it was about 4 °C at night and the temperature is usually stable in Ann Arbor.
3. I would use regression to predict the max temperature in Ann Arbor on October 31, 2026. That way we can predict the temperature based on recent years, accounting for climate change
4. It looks like the max temperature is around 22 C, found through mutating a lag column, filtering by month and day equaling halloween, and checking correlation

Name an occupation you expect to be highly exposed to AI. Name an occupation you expect to be highly resistant to AI.

77 anonymous responses

Software development seen as exposed	31
Responses that name software engineering, programming, or related development roles as occupations likely to be highly exposed to AI.	
Accounting and business roles seen exposed	11
Responses that identify accounting, business intelligence, or related analytical/office roles as highly exposed to AI.	
Explicit contrast: software exposed, manual resistant	9
Responses that state software developers/programmers are highly exposed and contrast them with manual, hands-on jobs (e.g., construction) as resistant due to physical tasks.	
AI developers labeled as exposed	7
Responses that identify AI developers or SDE working on AI itself as highly exposed to AI-related change.	

Response themes for question 2

Software exposed, medical profession resistant**6**

Responses that pair software engineering or programming as highly exposed with doctors/medical professions as resistant to AI.

Entry-level analytics/finance exposed**5**

Responses that specify entry-level data analytics, data scientist, or financial analyst roles as particularly exposed to AI.

Healthcare and mental health seen resistant**4**

Responses that name nursing, doctors, or psychological/therapy roles as resistant to AI due to complex human interaction and care.

Service or physical jobs seen resistant**4**

Responses that name service or hands-on roles (e.g., waiters, bread cutter, athlete) as resistant to AI.

Student responses

1. Accounting or low-level software programming would be highly exposed to AI, tsince models can reliably work with digital data.
2. Anything with coding is probably very exposed to AI and something really not exposed might be blue-collar jobs like police officers, construction workers.
3. coding works , cheifs
4. computer science I think is highly exposed while something like construction would be highly resistant
5. computer science that is just direct coding, and then highly resistant is blue collar labor.
6. CS is probably highly exposed to AI, and construction or blue collar jobs are probably highly resistant to AI.
7. highly exposed is a c.s job, highly resistant would be a human labor job, like construction
8. highly exposed would be entry-level software engineers. highly resistant would be doctors at the moment
9. I am the AI engineer, and elementary level of coders, t

Student responses

1. I expect computer science to be highly exposed to AI, and fields in the arts to be highly resistant to AI.
2. I expect Programming/ Software development to be exposed to AI and sales to be resistant to AI
3. I think programming is highly exposed, but nursing is very resistant
4. I think that jobs involving coding are highly exposed to AI while sports jobs are not
5. I'd expect jobs such as computer science and pure mathematics to be more easily replaced by AI jobs, whereas work like construction and farming to be a long way off.
6. One occupation I expect to be highly exposed to AI is software engineering/development, because of the availability of training data and nature of the job. One that wouldn't be exposed is physical labor.
7. One occupation I think is most exposed to AI is junior software developer, and an occupation I expect to be resistant to AI is plumbing.
8. Programmer
9. programmer? resistant maybe film director?

Student responses

1. Software development would be the most impacted, while medical work would be the least impacted
2. software development, sales
3. software engineer could be exposed. heart surgeon would not be
4. Software engineer would be highly exposed to AI, while a nurse or doctor may be highly resistant
5. Software engineer, chef
6. software engineer, plumber
7. software engineerign highly exposed to AI job \s like plubling is hard labour will be reiitant tol AI
8. Software engineering, arts and crafts
9. Software engineering, data entry, and data science jobs are highly exposed. Nursing, manual labor, and trades are highly resistant to AI

Student responses

1. Software engineering, maybe data science, perhaps finances, or something that does calculations. I would say other careers that aren't something that can be made predictions or models of are more safe?
2. SWE, construction workers
3. SWE, Fast food
4. Tech highly exposed, medicine highly resistant

Student responses

1. Accounting because I once majored in. Then should be therapy, mental care giver
2. Accounting is the most exposed, while manufacturing is more resistant.
3. An occupation that is highly exposed to AI could be along the lines of SWE, whereas a blue collar job like manufacturing I expect to be highly resistant to AI
4. Business Intelligence is very exposed, and waiters is not
5. highly exposed: sanitor resistant: teacher
6. I think data entry roles are highly exposed. Professions like nurses or dentists are not highly exposed
7. I think that a paralegal would be highly exposed to AI as a lot of their work is research while a construction worker is less exposed.
8. I think that an accountant would be highly perceptible to ai and i think carpenters would be not exposed to ai
9. labor jobs would be resistant while secretarial jobs would be exposed.

Student responses

1. One occupation would be early graphic design with text generating graphics. Data Entry and Coding is up there too
2. One profession that is exposed to AI is analytics, one that isnt as much is maybe coaching or teaching

Student responses

1. A service job directly interacting with customers like a food service job I'd think is resistant. A job like a middle manager might be easily replaced since it can be trained for
2. A software developer could be highly exposed to AI because AI can help with coding tasks, while a construction worker would be more resistant because the job requires physical, hands-on work.
3. computer science is exposed to AI, resistant would be construction
4. Computer science is highly exposed, but actors are highly resistant.
5. Hands on jobs like electricians, plumbers, etc would be resistant to AI because those can not be replaced, but software engineering or other tech jobs would be sensitive to AI
6. Highly exposed: Computer scientist/ computer engineer Resistant: Construction worker
7. i think computer science, because ai is able to do coding very well. teaching is highly resistant because live instruction cannot be replicated
8. Machine learning and software engineering jobs are highly exposed and blue color jobs like being a cop is highly resistant to AI

Student responses

1. Software development is highly exposed to AI, a trade like plumbing is not very exposed to AI

Student responses

1. AI developer and doctor
2. customer service roles are highly exposed to AI with many of these roles already replaced by AI
3. data science or software engineering have lots of AI. I would imagine management to be resistant to AI.
4. Designed-based and analytical/economic jobs may be highly exposed to AI. I believe that physical science, like research, may be resistant because humans are required to actually conduct experiments.
5. education
6. I think that the people who handle call centers and customer service and stuff to be mostly replaced by AI. I expect engineers to be resistant.
7. Physical jobs like construction.

Student responses

1. Highly exposed to AI: entry-level data-scientists. Highly resistant to AI: doctors
2. highly exposed: software engineer highly resistant: hairstylist
3. Highly resistant: nursing, exposed: anything to do with comp sci, software
4. I feel like doctor is a pretty AI resistant job
5. I think accounting would be high exposed to AI because you're inputting journal entries everyday and its super repeative. Doctors would be AI resistnat
6. Software Engineering, AI can basically do, will be vulnerable. Healthcare will be highly resistant because it is very personal.

Student responses

1. Data scientists, and lecturers or professors
2. entry level data analysis, resistant- confidential information/data/cybersecurity
3. Exposed: financial analysts Resistant: nursing
4. Highly exposed - entry level data analytics. Highly resistant - nursing.
5. Highly exposed to AI: Data analyst Highly resistant to AI: Plumber

Student responses

1. computer scientist is highly exposed; lead sport science technology assisant is resistant under the HPSSC program
2. I expect a data science position to be highly exposed to AI and a medical position highly resistant
3. I expect translators to be highly exposed because AI can help translate text. I expect plumbers to be more resistant because their work requires hands-on repairs in different physical settings.
4. SDE. A psychological doctor would be highly resistant to AI.

Student responses

1. anything office that is more menial (assistant jobs) would be vulnerable to AI, and trade jobs/in-person labor like electricians or plumbers are resistant
2. bread cutter will be resistant, consulting will be replaced with astra 6
3. exposed: data entry, repetitive work. resistant: hairdresser, doctor
4. related: SWE resistant: Athlete

- Join the tables by occupation code, keeping occupations found in both sources. Save the result as ai. - Find the five most AI-exposed occupations in the matched table. - Display their names and exposure scores.

80 anonymous responses

Other responses

80

Student responses

1. # Combine the sources, then find the highest exposure scores. `ai <- bls |> left_join(exposure, by = "occupation_code") ai |> arrange(desc(exposure_score)) |> slice_head(n = 5) |> select(occupation_name, exposure_score)`
2. # fold: [Match occupations across the two sources] `ai <- inner_join(bls, exposure, by = "occupation_code")` # endfold # fold: [Find the five most exposed occupations] `ai |> arrange(desc(exposure_percent)) |> select(occupation, exposure_percent) |> head(5)`
3. # fold: [Match occupations across the two sources] `ai <- inner_join(bls, exposure, by = "occupation_code")` # endfold # fold: [Find the five most exposed occupations] `ai |> arrange(desc(exposure_percent)) |> select(occupation, exposure_percent) |> head(5)` # Combine the sources, then find the highest exposure scores.

Student responses

1. # fold: [Match occupations across the two sources] ai <- inner_join(bls, exposure, by = "occupation_code") # endfold # fold: [Find the five most exposed occupations] ai |> arrange(desc(exposure_percent)) |> select(occupation, exposure_percent) |> head(5) # Combine the sources, then find the highest exposure scores.
2. # fold: [Match occupations across the two sources] ai <- inner_join(bls, exposure, by = "occupation_code") # endfold # fold: [Find the five most exposed occupations] ai |> arrange(desc(exposure_percent)) |> select(occupation, exposure_percent) |> head(5) # endfold
3. # fold: [Match occupations across the two sources] ai <- inner_join(bls, exposure, by = "occupation_code") # endfold # fold: [Find the five most exposed occupations] ai |> arrange(desc(exposure_percent)) |> select(occupation, exposure_percent) |> head(5) # endfold
4. ai <- bls %>% inner_join(exposure, by = "occupation_code") ai %>% arrange(desc(exposure_percent)) %>% select(occupation, exposure_percent) %>% slice_head(n = 5) # Combine the sources, then find the highest exposure scores.

Student responses

1. `ai <- bls |> inner_join(exposure, by = "occ_code") ai |> arrange(desc(exposure)) |> slice_head(n = 5) |> select(occupation, exposure) # Combine the sources, then find the highest exposure scores.`
2. `ai <- bls |> inner_join(exposure, by = "occupation_code") ai |> arrange(desc(exposure_percent)) |> select(occupation, exposure_percent) |> head(5) # Combine the sources, then find the highest exposure scores.`
3. `ai <- bls |> inner_join(exposure, by = "occupation_code") ai |> arrange(desc(exposure_percent)) |> select(occupation, exposure_percent) |> head(5) # Combine the sources, then find the highest exposure scores.`
4. `ai <- bls |> inner_join(exposure, by = "occupation_code") ai |> arrange(desc(exposure_percent)) |> select(occupation, exposure_percent) |> slice_head(n = 5)`
5. `ai <- bls |> inner_join(exposure, by = "occupation_code") ai |> arrange(desc(exposure_percent)) |> slice_head(n = 5) |> select(occupation_name, exposure_percent) # Combine the sources, then find the highest exposure scores.`

Student responses

1. `ai <- bls |> inner_join(exposure, by = "occupation_code") |> arrange(desc(exposure_percent)) |> slice_head(n = 5) #` Combine the sources, then find the highest exposure scores.
2. `ai <- bls |> inner_join(exposure, by = "occupation_code") |> arrange(desc(exposure_percent)) |> head(n=5) |> select(occupation, exposure_percent) ai #`
Combine the sources, then find the highest exposure scores.
3. `ai <- bls |> inner_join(exposure, by = "occupation_code") |> arrange(desc(exposure_percent)) |> slice_head(n = 5) ai`
4. `ai <- bls |> inner_join(exposure, by = "occupation_code") |> arrange(desc(exposure_percent)) |> slice_head(n = 5) |> select(occupation, exposure_percent) #`
Combine the sources, then find the highest exposure scores.
5. `ai <- bls |> inner_join(exposure, by = "occupation_code") |> arrange(desc(exposure_percent)) |> slice_head(n = 5) |> select(occupation, exposure_percent) #`
Combine the sources, then find the highest exposure scores.

Student responses

1. `ai <- bls |> inner_join(exposure, by = "occupation_code") |> arrange(desc(exposure_percent)) |> slice_head(n = 5) |> select(occupation, exposure_percent) ai #`
Combine the sources, then find the highest exposure scores.
2. `ai <- bls |> inner_join(exposure, by = "occupation_code") |> arrange(exposure_percent) |> slice_head(n = 5) ai #` Combine the sources, then find the highest exposure scores.
3. `ai <- bls |> inner_join(exposure, by='occupation_code') ai |> arrange(desc(exposure_percent)) |> slice_head(n=5) |> select(occupation, exposure_percent) #`
Combine the sources, then find the highest exposure scores.
4. `ai <- bls |> inner_join(exposure, by = "occupation_code") |> arrange(desc(exposure_percent)) |> slice_head(n = 5) #` Combine the sources, then find the highest exposure scores.
5. `ai <- bls |> inner_join(exposure, by = "occupation_code") |> arrange(desc(exposure_percent)) |> slice_head(n = 5) |> select(occupation, exposure_percent) #`
Combine the sources, then find the highest exposure scores.

Student responses

1. `ai <- bls |> inner_join(exposure, by = 'occupation_code') ai |> arrange(desc(exposure_percent)) |> slice_head(n = 5) |> select(occupation, exposure_percent) #`
Combine the sources, then find the highest exposure scores.
2. `ai <- bls |> inner_join( exposure, by = "occupation_code") |> arrange(desc(exposure_percent)) |> slice_head(n = 5) |> select(occupation, exposure_percent) ai #`
Combine the sources, then find the highest exposure scores.
3. `ai <- bls |> inner_join(exposure, by = "occupation_code") ai |> arrange(desc(exposure_percent)) |> slice_head(n = 5) |> select(occupation, exposure_percent) #`
Combine the sources, then find the highest exposure scores.
4. `ai <- bls |> inner_join(exposure, by = 'occupation_code') ai |> arrange(desc(exposure_percent)) |> head(5) #` Combine the sources, then find the highest exposure scores.
5. `ai <- bls |> inner_join(exposure, by = "occupation_code") ai |> arrange(desc(exposure_percent)) |> head(5) |> select(occupation, exposure_percent) #` Combine the sources, then find the highest exposure scores.

Student responses

1. `ai <- bls |> inner_join(exposure, by = "occupation_code") ai |> arrange(desc(exposure_percent)) |> select(occupation, exposure_percent) |> head(5) # Combine the sources, then find the highest exposure scores.`
2. `ai <- bls |> inner_join(exposure, by = "occupation_code") ai |> arrange(desc(exposure_percent)) |> slice_head(n = 5) # Combine the sources, then find the highest exposure scores.`
3. `ai <- bls |> inner_join(exposure, by = "occupation_code") ai |> arrange(desc(exposure_score)) |> select(occupation_title, exposure_score) |> head(5) # Combine the sources, then find the highest exposure scores.`
4. `ai <- bls |> inner_join(exposure, by = "occupation_code") ai |> arrange(desc(exposure_percent)) |> slice_head(n=5) |> select(occupation, exposure_percent) # Combine the sources, then find the highest exposure scores.`
5. `ai <- bls |> inner_join(exposure, by="occupation_code") ai |> arrange(desc(exposure_percent)) |> select(occupation, exposure_percent) |> slice_head(n=5) # Combine the sources, then find the highest exposure scores.`

Student responses

1. `ai <- bls |> inner_join(exposure) |> arrange(desc(exposure_percent)) |> slice_head(n = 5) ai |> select(occupation, exposure_percent, projected_growth) #`
Combine the sources, then find the highest exposure scores.
2. `ai <- bls |> left_join(exposure, by = "occupation_code") ai |> arrange(desc(exposure_percent)) |> select(occupation, exposure_percent) #` Combine the sources, then find the highest exposure scores.
3. `ai <- bls |> left_join(exposure, by = "occupation_code") > arrange(desc(exposure_percent)) |> select(occupation, exposure_percent) ai #` Combine the sources, then find the highest exposure scores.
4. `ai <- bls |> inner_join(exposure, by='occupation_code') |> arrange(desc(exposure_percent)) |> select(occupation, exposure_percent) |> head(n=5) ai #` Combine the sources, then find the highest exposure scores.
5. `ai <- exposure |> inner_join(bls, by = "occupation_code") ai |> arrange(desc(exposure_percent)) |> select(occupation, exposure_percent) |> head(5)`

Student responses

1. `ai <- inner_join(bls, exposure, by = "occ_code") ai |> slice_max(exposure_score, n = 5) |> select(occupation, exposure_score) # Combine the sources, then find the highest exposure scores.`
2. `ai <- inner_join(bls, exposure, by = "occupation_code"); top5 <- ai |> arrange(desc(exposure_percent)) |> head(5); top5 |> select(occupation, exposure_percent)`
3. `ai <- inner_join(exposure, bls, by = "occupation_code") ai |> arrange(desc(exposure_percent)) |> head(5) |> select(exposure_percent, occupation) # Combine the sources, then find the highest exposure scores.`
4. `ai <- bls |> left_join(exposure, "occupation_code") ai # Combine the sources, then find the highest exposure scores.`
5. `ai = bls |> inner_join(exposure, by = 'occupation_code') |> arrange(desc(exposure_percent)) |> select(occupation, exposure_percent) |> print() # Combine the sources, then find the highest exposure scores.`
6. `ai <- bls |> inner_join(exposure, by = "occupation_code") ai |> arrange(desc(exposure_percent)) |> head(5) # Combine the sources, then find the highest exposure scores.`

Student responses

1. `ai <- bls |> inner_join(exposure, by="occupation_code") |> arrange(desc(exposure_percent)) |> slice_head(n=5) # Combine the sources, then find the highest exposure scores.`
2. `ai <- bls |> left_join(exposure, by = "occupation_code") ai |> arrange(desc(exposure_percent)) |> head(5) # Combine the sources, then find the highest exposure scores.`
3. `bls ai <- bls |> inner_join(exposure, by = "occupation_code") ai |> arrange(desc(exposure_score)) |> slice_head(n = 5) |> select(occupation_name, exposure_score) # Combine the sources, then find the highest exposure scores.`
4. `bls ai <- bls |> inner_join(exposure, by = "occupation_code") |> arrange(desc(exposure_percent)) |> select(occupation, exposure_percent) |> head(5) # Combine the sources, then find the highest exposure scores.`
5. `bls ai <- inner_join(occ, exposure, by = "occupation_code") ai %>% select(occupation_name, exposure_score) %>% arrange(desc(exposure_score)) %>% slice_head(n = 5) # Combine the sources, then find the highest exposure scores.`

Student responses

1. `bls library(dplyr) library(dplyr) ai <- ai_exposure |> inner_join(bls, by = "occupation_code") top_5_exposed <- ai |> arrange(desc(exposure_score)) |> slice_head(n = 5) |> select(occupation, exposure_score) print(top_5_exposed)`
2. `bls <- inner_join(table1, table2, by = "occupation_code") top_5 <- bls %>% arrange(desc(exposure_score)) %>% slice_head(n = 5) %>% select(occupation_name, exposure_score) print(top_5) # Combine the sources, then find the highest exposure scores.`
3. `bls |> inner_join(exposure, by = "occupation_code") |> arrange(desc(exposure_percent)) |> slice_head(n = 5) |> select(occupation, exposure_percent) # Combine the sources, then find the highest exposure scores.`
4. `bls |> inner_join(exposure, by = "occupation_code") |> arrange(desc(exposure_percent)) |> # Combine the sources, then find the highest exposure scores.`

Student responses

1. `bls |> ai <- inner_join(exposure, occupations, by = "occ_code") ai_top5 <- ai |> arrange(desc(exposure_score)) |> select(occupation_name, exposure_score) |> slice_head(n = 5) ai_top5` # Combine the sources, then find the highest exposure scores.
2. `bls |> inner_join(exposure, by = "occupation_code") |> arrange(desc(exposure_percent)) |> slice_head(5) |> select(occupation, exposure_percent)`
3. `bls |> inner_join(exposure, by = "occupation_code") |> arrange(desc(exposure_percent))` # Combine the sources, then find the highest exposure scores.
4. `bls |> inner_join(exposure, by = "occupation_code") |> arrange(desc(exposure_percent)) |> head(5)` # Combine the sources, then find the highest exposure scores.
5. `bls |> inner_join(exposure, by = "occupation_code") |> arrange(desc(exposure_percent)) |> slice_head(n = 5)` # Combine the sources, then find the highest exposure scores.
6. `bls |> inner_join(exposure, by = "occupation_code") |> arrange(desc(exposure_percent)) |> slice_head(n = 5) |> select(occupation, exposure_percent)` # Combine the sources, then find the highest exposure scores.

Student responses

1. `bls |> inner_join(exposure, by = "occupation_code") |> arrange(desc(exposure_percent)) |> slice_head(n = 5) |> select(occupation, exposure_percent) #`
Combine the sources, then find the highest exposure scores.
2. `bls |> inner_join(exposure, by = "occupation_code") |> arrange(desc(exposure_percent)) |> slice_head(n=5) |> select(occupation, exposure_percent) #`
Combine the sources, then find the highest exposure scores.
3. `bls |> inner_join(exposure, by = "occupation_code") |> arrange(desc(exposure_percent)) |> slice_head(n=5) |> select(occupation, exposure_percent) #`
Combine the sources, then find the highest exposure scores.
4. `bls |> inner_join(exposure, by = "occupation_code")|> arrange(desc(exposure_percent))|> slice_head(n=5) select(name,exposure_percent) #` Combine the sources, then find the highest exposure scores.
5. `bls |> inner_join(exposure, by = "occupation_code") |> arrange(desc(exposure_percent)) #` Combine the sources, then find the highest exposure scores.

Student responses

1. `bls |> inner_join(exposure, by = "occupation_code") |> arrange(desc(exposure_percent)) |> slice_head(n = 5) |> select(occupation, exposure_percent) #`
Combine the sources, then find the highest exposure scores.
2. `bls |> inner_join(exposure, by = "occupation_code") |> arrange(desc(exposure_percent)) |> slice_head(n = 5) ai |> arrange(desc(exposure_percent)) |>`
`select(occupation, exposure_percent) |> head(5) #` Combine the sources, then find the highest exposure scores.
3. `bls |> inner_join(exposure, by = "occupation_code") |> arrange(desc(exposure_percent)) |> head(5) |> select(occupation, exposure_percent) #` Combine the
sources, then find the highest exposure scores.
4. `bls |> inner_join(exposure, by = "occupation_code") |> arrange(desc(exposure_percent)) |> select(occupation, exposure_percent) |> head(5) #` Combine the
sources, then find the highest exposure scores.

Student responses

1. `bls |> inner_join(exposure, by = "occupation_code") |> arrange(desc(exposure_percent)) |> slice_head(n = 5) |> select(occupation, exposure_percent) #`
Combine the sources, then find the highest exposure scores.
2. `bls |> inner_join(exposure, by = "occupation_code") |> arrange(desc(exposure_percent)) |> slice_head(n = 5) |> select(occupation, exposure_percent) #`
Combine the sources, then find the highest exposure scores.
3. `bls |> inner_join(exposure, by = "occupation_code") |> arrange(desc(exposure_percent)) |> slice_head(n = 5) |> select(occupation, exposure_percent)`
4. `bls |> inner_join(exposure, by = "occupation_code") -> ai ai |> arrange(desc(exposure_percent)) |> select(occupation, exposure_percent) |> slice_head(n=5) #`
Combine the sources, then find the highest exposure scores.
5. `bls |> inner_join(exposure, by = "occupation_code") |> arrange(desc(exposure_percent)) |> slice_head(n = 5) |> select(exposure_percent, occupation) #`
Combine the sources, then find the highest exposure scores.

Student responses

1. `bls |> inner_join(exposure, by = "occupation_code") |> arrange(desc(exposure_percent)) |> slice_head(n = 5) # fold: [Find the five most exposed occupations] ai
|> arrange(desc(exposure_percent)) |> select(occupation, exposure_percent) |> head(5) # Combine the sources, then find the highest exposure scores.`
2. `bls |> inner_join(exposure, by = "occupation_code") |> arrange(desc(exposure_percent)) |> select(occupation, exposure_percent) |> head(5) # Combine the
sources, then find the highest exposure scores.`
3. `bls |> inner_join(exposure, by = "occupation_code") |> arrange(desc(exposure_percent)) |> slice_head(n=5) |> select(occupation, exposure_percent) # Combine
the sources, then find the highest exposure scores.`
4. `bls |> inner_join(exposure, by= "occupation_code") |> arrange(desc(exposure_percent)) |> slice_head(n=5) |> select(occupation, exposure_percent)`
5. `bls |> left_join(exposure, by = "occupation_code") |> arrange(desc(exposure_percent)) # Combine the sources, then find the highest exposure scores.`

Student responses

1. `bls |> left_join(exposure,by="occupation_code") |> arrange(desc(exposure_percent))` # Combine the sources, then find the highest exposure scores.
2. `bls|> inner_join(exposure,by = "occupation_code") arrange(desc())` # Combine the sources, then find the highest exposure scores.
3. `bls|>inner_join(exposure, by = "occupation_code")|> arrange((exposure_percent))|> select(occupation, exposure_percent)` # Combine the sources, then find the highest exposure scores.

- Rank all occupations in ai by exposure. Give the highest value rank 1, with equal ranks for ties. - Keep the row for Data scientists. - Display the occupation name, exposure score, and rank.

75 anonymous responses

Other responses

75

Student responses

1. ai # Find the exposure rank for data scientists.
2. ai ai |> mutate(rank = min_rank(desc(exposure))) |> filter(occupation == "Data scientists") |> select(occupation, exposure, rank) # Find the exposure rank for data scientists.
3. ai # Find the exposure rank for data scientists. ai_ranked <- ai |> mutate(rank = min_rank(desc(exposure_score))) data_scientist_rank <- ai_ranked |> filter(stringr::str_detect(occupation, "(?i)data scientist")) |> select(occupation, exposure_score, rank) print(data_scientist_rank)
4. ai ai |> mutate(rank = min_rank(desc(exposure_score))) |> filter(occupation == "Data scientists") |> select(occupation, exposure_score, rank) # Find the exposure rank for data scientists.
5. ai |> mutate(exposure_rank = min_rank(desc(exposure_percent))) |> select(occupation, exposure_rank) |> filter(occupation == "Data scientists") |> select(occupation, exposure_percent, rank) # Find the exposure rank for data scientists.

Student responses

1. `ai %>% mutate(rank = min_rank(desc(exposure))) %>% filter(`OCC_TITLE` == "Data scientists") %>% select(OCC_TITLE, exposure, rank) # Find the exposure rank for data scientists.`
2. `ai |> # fold: [Rank all occupations by exposure] mutate(exposure_rank = min_rank(desc(exposure_percent))) |> # endfold # fold: [Find data scientists] filter(occupation == "Data scientists") |> select(occupation, exposure_percent, exposure_rank) # endfold`
3. `ai |> # fold: [Rank all occupations by exposure] mutate(exposure_rank = min_rank(desc(exposure_percent))) |> # endfold # fold: [Find data scientists] filter(occupation == "Data scientists") |> select(occupation, exposure_percent, exposure_rank) # endfold`
4. `ai |> mutate(exposure_rank = min_rank(desc(exposure_percent))) |> filter(occupation == 'Data scientists') |> select(occupation, exposure_percent, exposure_rank) # Find the exposure rank for data scientists.`

Student responses

1. `ai |> mutate(exposure_rank = min_rank(desc(exposure_percent))) |> arrange(exposure_rank) |> select(occupation, exposure_rank, exposure_percent) |> filter(occupation == 'Data scientists') # Find the exposure rank for data scientists.`
2. `ai |> mutate(exposure_rank = min_rank(desc(exposure_percent))) |> filter(occupation == "data scientists") |> select(occupation, exposure_rank, rank) # Find the exposure rank for data scientists.`
3. `ai |> mutate(exposure_rank = min_rank(desc(exposure_percent))) |> arrange(exposure_rank) |> select(occupation, exposure_percent, exposure_rank) |> filter(occupation == "Teacher") # Find the exposure rank for data scientists.`
4. `ai |> mutate(exposure_rank = min_rank(desc(exposure_percent))) |> arrange(exposure_rank) |> select(occupation, exposure_rank, exposure_percent) |> filter(occupation == "Data scientists") # Find the exposure rank for data scientists.`

Student responses

1. `ai |> mutate(exposure_rank = min_rank(desc(exposure_percent))) |> arrange(exposure_rank) |> select(occupation, exposure_rank, exposure_percent) |> filter(occupation == "Data scientists") # Find the exposure rank for data scientists.`
2. `ai |> mutate(exposure_rank = min_rank(desc(exposure_percent))) |> arrange(exposure_rank) |> select(occupation, exposure_rank, exposure_percent) |> filter(occupation == "Data scientists") # Find the exposure rank for data scientists.`
3. `ai |> mutate(exposure_rank = min_rank(desc(exposure_percent))) |> arrange(exposure_rank) |> select(occupation, exposure_rank) |> filter(occupation == "Data scientist") # Find the exposure rank for data scientists.`
4. `ai |> mutate(exposure_rank = min_rank(desc(exposure_percent))) |> arrange(exposure_rank) |> select(occupation, exposure_rank) |> filter(occupation == "Data scientists") |> select(occupation, exposure_percent, rank) # Find the exposure rank for data scientists.`

Student responses

1. `ai |> mutate(exposure_rank = min_rank(desc(exposure_percent))) |> filter(occupation == "business") |> select(occupation, exposure_percent, exposure_rank)`
2. `ai |> mutate(exposure_rank = min_rank(desc(exposure_percent))) |> filter(occupation == "Data scientists") |> select(occupation, exposure_percent, exposure_rank)`
3. `ai |> mutate(exposure_rank = min_rank(desc(exposure_percent))) |> filter(occupation == "Data scientists") |> select(occupation, exposure_percent, exposure_rank)`
4. `ai |> mutate(exposure_rank = min_rank(desc(exposure_percent))) |> filter(occupation == "Data scientists") |> select(occupation, exposure_percent, exposure_rank) # Find the exposure rank for data scientists.`
5. `ai |> mutate(exposure_rank = min_rank(desc(exposure_percent))) |> filter(occupation == "Data scientists") |> select(occupation, exposure_percent, exposure_rank) # Find the exposure rank for data scientists.`

Student responses

1. `ai |> mutate(exposure_rank = min_rank(desc(exposure_percent))) |> filter(occupation == "Data scientists") |> select(occupation, exposure_percent, rank) #`
Find the exposure rank for data scientists.
2. `ai |> mutate(exposure_rank = min_rank(desc(exposure_percent))) |> filter(occupation == "Data scientists") |> select(occupation, exposure_percent, rank)`
`head(3) #` Find the exposure rank for data scientists.
3. `ai |> mutate(exposure_rank = min_rank(desc(exposure_percent))) |> filter(occupation == "Data scientists") |> select(occupation, exposure_rank,`
`exposure_percent)`
4. `ai |> mutate(exposure_rank = min_rank(desc(exposure_percent))) |> select(occupation, exposure_percent, exposure_rank) |> filter(occupation == "Data`
`scientists") #` Find the exposure rank for data scientists.
5. `ai |> mutate(rank = min_rank(desc(exposure_percent))) |> arrange(rank) |> filter(occupation == "Data scientists") |> select(occupation, exposure_percent,`
`rank) #` Find the exposure rank for data scientists.

Student responses

1. `ai |> mutate(rank = min_rank(desc(exposure_percent))) |> filter(occupation == "Data scientists") |> select(occupation, exposure_percent, rank) # Find the exposure rank for data scientists.`
2. `ai |> mutate(rank = min_rank(desc(exposure_percent))) |> filter(occupation == "Data scientists") |> select(occupation, exposure_percent, rank) # Find the exposure rank for data scientists.`
3. `ai |> mutate(rank = min_rank(desc(exposure_percent))) |> filter(occupation == "Data Scientists") |> select(occupation, exposure_percent, rank) # Find the exposure rank for data scientists.`
4. `ai |> mutate(rank = min_rank(desc(exposure_percent))) |> filter(occupation == "Data scientists") |> select(occupation, rank, exposure_percent) ai # Find the exposure rank for data scientists.`
5. `ai |> mutate(rank = min_rank(desc(exposure_percent))) |> filter(occupation == "Data scientists") |> select(occupation, exposure_percent, rank) # Find the exposure rank for data scientists.`

Student responses

1. ai |> mutate(rank = min_rank(desc(exposure_percent))) |> select(occupation, rank) |> filter(occupation == "Data scientists") |> select(occupation, exposure_percent, rank) # Find the exposure rank for data scientists.
2. ai |> mutate(rank = min_rank(desc(exposure_score))) |> filter(occupation == "Data scientists") |> select(occupation, exposure_score, rank) # Find the exposure rank for data scientists.
3. ai |> mutate(rank = min_rank(desc(exposure_score))) |> filter(occupation_name == "Data Scientists") |> select(occupation_name, exposure_score, rank) # Find the exposure rank for data scientists.
4. ai |> mutate(rank = min_rank(desc(exposure_score))) |> filter(occupation_title == "Data scientists") |> select(occupation_title, exposure_score, rank) # Find the exposure rank for data scientists.
5. ai |> mutate(rank = min_rank(desc(exposure))) |> filter(occupation == "Data Scientists") |> select(occupation, exposure, rank) # Find the exposure rank for data scientists.

Student responses

1. `ai |> mutate(rank=min_rank(desc(exposure_percent))) filter(occupation=="Data_scientist") select(occupation,rank)|> # Find the exposure rank for data scientists.`
2. `ai |> mutate(exposure_rank = min_rank(desc(exposure_percent))) |> filter(occupation == "Data scientists") |> select(occupation, exposure_rank, rank) # Find the exposure rank for data scientists.`
3. `ai |> mutate( exposure_rank = min_rank(desc(exposure_percent)) ) |> filter(occupation = "Data scientists") |> select(occupation, exposure_percent, rank)`
4. `ai |> mutate(exposure_rank = min_rank(desc(exposure_percent))) |> filter(occupation == "Data scientists") |> select(occupation, exposure_percent, exposure_rank) # Find the exposure rank for data scientists.`
5. `ai |> mutate(exposure_rank = min_rank(desc(exposure_percent))) |> filter(occupation == "Data scientists") |> select(occupation, exposure_percent, exposure_rank) # Find the exposure rank for data scientists.`

Student responses

1. `ai |> mutate(exposure_rank = min_rank(desc(exposure_percent))) |> filter(occupation == "Data scientists") |> select(occupation, exposure_percent, exposure_rank) |> slice_head(n=5) # Find the exposure rank for data scientists.`
2. `ai |> mutate(exposure_rank = min_rank(desc(exposure_percent))) |> filter(occupation == "Psychologists") |> select(occupation, exposure_rank, exposure_percent) # Find the exposure rank for data scientists.`
3. `ai |> mutate(exposure_rank = min_rank(desc(exposure_percent))) |> select(occupation, exposure_rank, exposure_score) |> filter(occupation == "Data scientists") # Find the exposure rank for data scientists.`
4. `ai |> mutate(rank = min_rank(desc(exposure_percent))) # Find the exposure rank for data scientists.`
5. `ai |> mutate(rank = min_rank(desc(exposure_percent))) |> filter(occupation == "Data scientists") |> select(occupation, exposure_percent, rank) # Find the exposure rank for data scientists.`

Student responses

1. `ai |> mutate(rank = min_rank(desc(exposure_percent))) |> filter(occupation == "Data scientists") |> select(occupation, exposure_percent, rank) # Find the exposure rank for data scientists.`
2. `ai |> mutate(rank = min_rank(desc(exposure_percent))) |> filter(occupation == "Data scientists") |> select(occupation, exposure_percent, rank) # Find the exposure rank for data scientists.`
3. `ai |> mutate(rank = min_rank(desc(exposure_percent))) |> filter(occupation == "Data scientists") |> select(occupation, exposure_percent, rank) # Find the exposure rank for data scientists.`
4. `ai |> mutate(rank = min_rank(desc(exposure_percent))) |> filter(occupation == "Data scientists")|> select(occupation, rank, exposure_percent) # Find the exposure rank for data scientists.`
5. `ai |> mutate(rank=min_rank(desc(exposure_percent))) filter(occupation == "Data Scientists") |> select(occupation,exposure_percent, rank) # Find the exposure rank for data scientists.`

Student responses

1. `ai |> mutate(rank = min_rank(desc(exposure_percent))) |> filter(occupation == "Data scientists") |> select(occupation, exposure_percent, rank) # Find the exposure rank for data scientists.`
2. `ai |> mutate(rank = min_rank(desc(exposure_percent))) |> filter(occupation == "Data scientists") |> select(occupation, exposure_percent, rank) # Find the exposure rank for data scientists.`
3. `ai |> mutate(rank = min_rank(desc(exposure_percent))) |> select(occupation, rank, exposure_percent) |> filter(occupation == "Data scientists") # Find the exposure rank for data scientists.`
4. `ai |> mutate(rank = min_rank(desc(exposure_percent))) |> filter(occupation == "Data scientists") |> select(occupation, exposure_percent, rank) # Find the exposure rank for data scientists.`
5. `ai |> mutate(rank = min_rank(desc(exposure_percent))) |> filter(occupation == "Data scientists") |> select(occupation, exposure_percent, rank) # Find the exposure rank for data scientists.`

Student responses

1. ai |> mutate(exposure_rank = min_rank(desc(exposure_percent))) |> filter(occupation == "Data scientists") |> select(occupation, exposure_percent, exposure_rank)
2. ai |> mutate(rank = min_rank(desc(exposure_percent))) |> select(occupation, rank, exposure_percent) |> filter(occupation == "Data scientists") # Find the exposure rank for data scientists.
3. ai |> mutate(rank = min_rank(desc(exposure_percent))) |> arrange(rank) # Find the exposure rank for data scientists.
4. ai |> mutate(rank = min_rank(desc(exposure_percent))) |> filter(occupation == "Data scientists") # Find the exposure rank for data scientists.
5. ai |> mutate(rank = min_rank(desc(exposure_percent))) |> filter(occupation == "Data scientists") |> select(occupation, exposure_percent, rank) # Find the exposure rank for data scientists.
6. ai |> mutate(rank = min_rank(desc(exposure_percent))) |> select(occupation, rank) # Find the exposure rank for data scientists.

Student responses

1. `ai |> mutate(rank = min_rank(desc(exposure_percent))) |> select(occupation, rank) |> filter(occupation == "Data scientists") |> select(occupation, exposure_percent, rank) # Find the exposure rank for data scientists.`
2. `ai |> mutate(rank = min_rank(desc(exposure_percent))) |> select(occupation, rank) |> filter(occupation == "Data scientists") |> select(rank, occupation, exposure_percent) # Find the exposure rank for data scientists.`
3. `ai |> mutate(rank = min_rank(desc(exposure_percent))) |> filter(occupation == "Data scientists") |> select(occupation, exposure_percent, rank) # Find the exposure rank for data scientists.`
4. `ai |> mutate(rank = min_rank(desc(exposure_percent)))|> select(occupation, rank, exposure_percent) |> filter(occupation == "Data scientists") # Find the exposure rank for data scientists.`
5. `ai |> mutate(rank = min_rank(desc(exposure_percent)))|> select(occupation,rank)|> filter(occupation == "Data scientists")|> select(occupation, exposure_persent, rank) # Find the exposure rank for data scientists.`

Student responses

1. `ai |> mutate(rank=min_rank(desc(exposure_percent))) |> select(occupation, rank, exposure_percent) |> filter(occupation == "Data scientists") # Find the exposure rank for data scientists.`
2. `ai|> mutate(exposure_rank = min_rank(desc(exposure_percent))) |> filter(occupation == "Data scientists") select(occupation, exposure_percent,rank) # Find the exposure rank for data scientists.`
3. `ai|> mutate(rank = min_rank(desc(exposure_percent)))|> select(occupation, rank, exposure_percent)|> filter(occupation == "Data scientists") # Find the exposure rank for data scientists.`
4. `ai|> mutate(rank = min_rank(exposure_percent))|> filter(occupation == "Data sciencists")|> select(occupation, exposure_percent,rank) # Find the exposure rank for data scientists.`
5. `ai|>mutate(rank = min_rank(exposure_percent))|> select(occupation, rank)|> filter(occupation == "Data Scientists")|> select(occupation, exposure_percent,rank) # Find the exposure rank for data scientists.`

Student responses

1. `ds <- ai |> mutate(exposure_rank = min_rank(desc(exposure_percent))) |> arrange(exposure_rank) |> filter(occupation == "Data scientists") |> select(occupation, exposure_percent, exposure_rank) ds`
2. `ranked_occupations <- ai %>% mutate(rank = min_rank(desc(exposure_score))) |> filter(occupation == "Data scientists") |> select(occupation, exposure_score, rank) print(ranked_occupations)`

Would you expect highly exposed occupations to grow faster or slower? Explain how AI could lead to the outcome you predict.

81 anonymous responses

Replacement causes slower growth

27

Asserts slower growth specifically because AI may replace much of the manual or routine work.

Automation reduces labor demand

22

States that AI will automate tasks within exposed occupations, leading firms to need fewer workers and causing slower growth.

Adaptation leads to faster growth

7

Argues that people and firms will adapt to AI, enabling more work and causing occupations to grow faster.

Occupation is threatened

7

Expresses a concise view that highly exposed occupations face threat or decline from AI without further elaboration.

Response themes for question 5

Reduced employment opportunities	7
Highlights that AI exposure leads to fewer job openings or lower employment prospects, often referencing a high exposure score.	
Role changes, slight shrinkage expected	5
Says occupations might shrink somewhat but also change in duties rather than disappear.	
Single-phrase shrinkage claim	4
Very short responses indicating exposed occupations will shrink or decline, without added explanation.	
Other responses	2

Student responses

1. could mean ai could have help data scientist
2. Faster
3. Faster, because their job is exposed to ai.
4. grow slower. AI could lead to this outcome because companies would feel less reluctant to hire people for automatable work
5. Highly exposed fields will probably have less people working in them.
6. Highly exposed occupations would expand slower, meaning people cannot get jobs.
7. Highly exposed occupations would probably grow faster because the AI will make the work more efficient so people will work faster and accomplish more
8. I expect highly exposed occupations to grow slower (reach a plateaued sort of growth pattern). AI could either accelerate current work, or replace people.
9. I expect that these occupations to grow faster

Student responses

1. I think highly exposed occupations would decline because it might replace workforce
2. I think I would expect them to grow much slower as since they are exposed to AI, then there would be a less reason to want to hire data scientists. AI would essentially removed more workers
3. I think it will cause the occupations to grow faster, if AI is able to replace some competency and make more people effective, jobs will increase with lower wages
4. I think it will make it grow faster because AI interest is growing.
5. I think it would be constant because AI can help data scientists be better. I expect more upper level roles and fewer entry level.
6. I think the AI could lead the unemployment of data analyst, but the job occupation for data scientist would increase
7. I would expect exposed careers to shrink as there is less demand and, therefore, less people hiring
8. I would expect exposed occupations to shrink at first, but when it adjusts it will grow faster.

Student responses

1. I would expect highly exposed fields to grow slower than unexposed occupations. If a field is exposed to AI, it decreases the amount of low-level positions are available.
2. I would expect highly exposed occupations to grow more slowly. If AI can perform a large share of an occupation's tasks, employers may need fewer workers to produce the same output, so hiring in those roles could slow down. That said, AI could also make those workers much more productive, which might expand demand for what they produce and support growth, so the net effect depends on whether the substitution or the productivity effect dominates.
3. I would expect highly exposed occupations to initially exponentially grow, but not grow from a human standpoint, but then when the bubble pops, it would shrink massively
4. I would expect occupations to decline
5. I would expect some highly AI-exposed occupations to grow faster because AI may increase worker productivity rather than completely replace workers. If AI allows workers to perform tasks more efficiently, demand for workers who can effectively use AI could increase.

Student responses

1. I would expect them to grow more slowly since the people in the field would keep their jobs and do less work while entry level jobs wont become open since they can just use ai
2. I would expect them to replace some jobs while also helping employees
3. slower
4. Slower, ai may replace a lot of the mduane ework
5. Slower? because most people wont join in a bad future job

Student responses

1. grow slower since companies will try to replace them w ai
2. Highly exposed occupations would probably grow slower as new analysts jobs are continuously replaced by ai.
3. I expect highly exposed occupations to shrink because AI can automate processes that may have required multiple employees previously.
4. I expect these occupations to grow more slowly, as AI would likely be used in the place of hiring new workers to do the same jobs.
5. I would expect AI exposed occupations to grow slower. How AI exposed it is could predict how little it will grow.
6. I would expect fields that are highly exposed to have slower growth rates since much of the work can be automated by AI
7. I would expect highly exposed occupations to grow slower because AI could automate some of their tasks, so companies may need fewer workers to do the same amount of work.
8. I would expect highly exposed occupations to grow slower because there will be less of a need for people in these positions, with AI having the capability to replace positions

Student responses

1. I would expect it to grow slower since AI can do many jobs that humans once did, making it more efficient and cost effective
2. I would expect the number of individuals employed to shrink, as AI takes over some of the work they do.
3. I would expect then to grow slower because there weren't be as many entry roles and people might not be leaving the industry if they already have a job.
4. I would expect these highly exposed occupations to grow slower because companies will begin implementing AI to budget cut, but it won't fully replace anything.
5. I'd assume that data scientist jobs will shrink in necessity if its exposed.
6. If most of the job can be done by AI then there's less demand for employees to do that job, so the occupations would grow slower.
7. One would expect shrinkage as AI could automate many tasks
8. slower because Ai will play a role in how much work they will do
9. Slower, because more of their jobs will be replaced by AI

Student responses

1. slower, or even shrink, since the jobs could be cut in favor of companies using AI
2. The existence of AI would lead those highly exposed occupations to grow slower as AI would gradually substitute their jobs.
3. This would mean that the hiring for the occupation would be at a much slower rate than for lesser-impacted positions
4. Yes. Lots of the tasks will be replaced by AI so people will hire less
5. You would expect occupations that are highly exposed to AI to grow slower, as new hires will have their jobs replaced.

Student responses

1. faster because as AI becomes more capable, they could do more in this areas
2. Faster, I'd expect people to adapt and for more work to be able to be done
3. Grow faster since they can be accelerated by AI
4. I think they would grow faster because of how AI can accelerate those occupations.
5. I would expect highly exposed AI occupations are expected to grow, thsi will enhance mundane task accomplishmnet speed and allows for more in-depth/better overall results
6. they would grow faster as the work will be more efficient
7. To grow faster, because people who are in these fields would keep trying harder to make themselves not replaced by AI

Student responses

1. Grow faster
2. Grow slower, since a lot of jobs would be easier with AI rather than people.
3. I expect highly exposed occupations will grow slower because their jobs will begin to be overtaken by artificial intelligence.
4. I expect that highly exposed occupations to shrink slower
5. I would expect highly exposed occupations to shrink on average. While some exposed professions will grow, some will also be automated away.
6. I would expect them to grow slower
7. That means that they are very threatened.

Student responses

1. I think AI will slow hiring so those occupations will go away, but also creative destruction so more jobs will be made eventually
2. I think this might mean that employment would go down because of the reliance of AI creating a lower need for workers
3. I would expect highly exposed occupations to grow slower since AI would maybe be taking over the entry level jobs in these highly exposed fields.
4. Less employment opportunity with a pretty high score
5. Probably grow slower because this occupation might not be in super high demand.
6. slower because theres less of a need for them
7. These occupations will have fewer growth because existing employees can be more efficient and can accomplish more.

Student responses

1. exposed lead to slower growth of those occupations by people and faster for machine automation
2. I think that highly exposed occupations that have a STEM background may actually have a higher growth rate. AI could help to improve efficiency in the work and draw forward innovation.
3. It might shrink a little bit, however possible not because it could just change what the position actually has to do
4. no idea
5. Not necessarily but I think the outcomes of the workers in these fields will be higher

Student responses

1. grow slower since ai will take over
2. highly exposed occupations will grow faster because there is more to do in the field regarding it
3. i would expect exposed fields to shrink
4. The exposed occupations may grow slower, it can replace

Use ``ai`` to make a scatterplot of projected growth against AI exposure, with one point per occupation. Label the axes. Run your code, then submit your plot. We will look at some of the plots together.

59 anonymous responses

Other responses

59

Student responses

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.
- 7.
- 8.
- 9.

Student responses

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.
- 7.
- 8.
- 9.

Student responses

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.
- 7.
- 8.
- 9.

Student responses

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.
- 7.
- 8.
- 9.

Student responses

- 1.
- 2.
- 3.
- 4.
5. `ai |> ggplot() + geom_point(aes( x = exposure_percent, y = projected_growth ))`
6. `ai |> ggplot() + geom_point(aes( x = exposure_percent, y = projected_growth))`
7. `ai |> ggplot() + geom_point(aes(x = exposure_percent, y = expected_growth))` # Plot projected growth against AI exposure.
8. `ai |> ggplot() + geom_point(aes(x = exposure_percent, y = projected_growth), alpha = .1) + labs(title = "Projected Growth vs Exposure Percent", x = "Exposure Percent", y = "Projected Growth"`
9. `ai |> ggplot() + geom_point(aes(x = exposure_percent, y = projected_growth))`

Student responses

1. `ai |> ggplot() + geom_point(aes(x = exposure_percent, y = projected_growth))`
2. `ai |> ggplot() + geom_point(aes(x = exposure_percent, y = projected_growth)) + labs( x = "AI exposure", y = "Projected growth" )`
3. `ai |> ggplot() + geom_point(aes(x = exposure_percent, y = projected_growth)) + labs( x = "Exposure Score", y = "Projected Growth" )`
4. `ai|> ggplot()+ geom_point(aes( x=exposure_percent, y=projected_growth))`
5. Here's my plot
6. I added a trend line as well as transparency for some dots and reduced their size.
7. i will add a title and axes labels
8. made dots smaller
9. made it more clear

Student responses

1. Scatterplot of projected employment growth against AI exposure score, one point per occupation. The cloud of points slopes slightly downward, which suggests that more AI-exposed occupations tend to have lower projected growth.
2. Seem to have a negative relationship between growth and exposure, but there are also outliers.
3. The cloud is mostly a shapeless blob with at best a slight downward lean, which is the point: exposure barely predicts growth.
4. The plot does not show a strong relationship, since jobs with higher AI exposure can still have either positive or negative projected growth.
5. We changed the dot little bit more transparent, so it would be more clear to see how the graph are scatter

- Keep occupations with exposure scores of at least 75. - For each family, make a boxplot, and add dots for the individual occupations.

77 anonymous responses

Other responses

77

Student responses

1. ai # Which kinds of highly exposed work are projected to grow?
2. ai # Which kinds of highly exposed work are projected to grow? ai_high_exposure <- ai |> filter(ai_score >= 75) ggplot(ai_high_exposure, aes(x = occupation_family, y = ai_score)) + geom_boxplot(outlier.shape = NA, fill = "gray95", color = "darkgray") + geom_jitter(width = 0.2, alpha = 0.6, color = "steelblue") + labs(title = "AI Exposure Distribution by Occupational Family (Score >= 75)", x = "Occupational Family", y = "AI Exposure Score") + theme_minimal()
3. ai high_exposure <- ai |> filter(exposure_percent >= 75) high_exposure |> ggplot(aes(x = family, y = exposure_score)) + geom_boxplot() + geom_jitter(width = 0.15, alpha = 0.6) # Which kinds of highly exposed work are projected to grow?

Student responses

1. `ai ai |> filter(exposure_percent >= 75) |> ggplot(aes(x = family, y = exposure_percent)) + geom_boxplot(outlier.shape = NA) + geom_jitter(width = 0.2, alpha = 0.6) + labs(x = "Occupation family", y = "AI exposure (%)") + theme_minimal() + theme(axis.text.x = element_text(angle = 45, hjust = 1)) # Which kinds of highly exposed work are projected to grow?`
2. `ai high_exposure <- occupations |> filter(exposure_score >= 75) ggplot(high_exposure, aes(x = family, y = exposure_score)) + geom_boxplot() + geom_jitter(width = 0.2, alpha = 0.6) + coord_flip() # Which kinds of highly exposed work are projected to grow?`
3. `ai %>% filter(exposure >= 75) %>% ggplot(aes(x = family, y = exposure)) + geom_boxplot() + geom_jitter(width = 0.2, alpha = 0.7) + labs(x = "Occupation family", y = "Exposure score", title = "Highly exposed occupations by family") + theme(axis.text.x = element_text(angle = 45, hjust = 1)) # Which kinds of highly exposed work are projected to grow?`

Student responses

1. `ai |> # fold: [Keep the highly exposed occupations] filter(exposure_percent >= 75) |> # endfold ggplot(aes(projected_growth, family)) + # fold: [Mark zero and draw one boxplot per family] geom_vline(xintercept = 0, color = "gray60") + geom_boxplot(outlier.shape = NA, width = 0.4) + # endfold # fold: [Show every occupation] geom_jitter(aes(color = family), width = 0, height = 0.08, size = 2) + # endfold # fold: [Label the plot] labs(x = "Projected employment growth, 2025–35 (%)", y = NULL) + theme(legend.position = "none", axis.text.y = element_text(size = 12)) # Which kinds of highly exposed work are projected to grow?`

Student responses

1. `ai |> # fold: [Keep the highly exposed occupations] filter(exposure_percent >= 75) |> # endfold ggplot(aes(projected_growth, family)) + # fold: [Mark zero and draw one boxplot per family] geom_vline(xintercept = 0, color = "gray60") + geom_boxplot(outlier.shape = NA, width = 0.4) + # endfold # fold: [Show every occupation] geom_jitter(aes(color = family), width = 0, height = 0.08, size = 2) + # endfold # fold: [Label the plot] labs(x = "Projected employment growth, 2025–35 (%)", y = NULL) + theme(legend.position = "none", axis.text.y = element_text(size = 12)) # endfold`

Student responses

1. `ai |> # fold: [Keep the highly exposed occupations] filter(exposure_percent >= 75) |> # endfold ggplot(aes(projected_growth, family)) + # fold: [Mark zero and draw one boxplot per family] geom_vline(xintercept = 0, color = "gray60") + geom_boxplot(outlier.shape = NA, width = 0.4) + # endfold # fold: [Show every occupation] geom_jitter(aes(color = family), width = 0, height = 0.08, size = 2) + # endfold # fold: [Label the plot] labs(x = "Projected employment growth, 2025–35 (%)", y = NULL) + theme(legend.position = "none", axis.text.y = element_text(size = 12)) # endfold`

Student responses

1. `ai |> # fold: [Keep the highly exposed occupations] filter(exposure_percent >= 75) |> # endfold ggplot(aes(projected_growth, family)) + # fold: [Mark zero and draw one boxplot per family] geom_vline(xintercept = 0, color = "gray60") + geom_boxplot(outlier.shape = NA, width = 0.4) + # endfold # fold: [Show every occupation] geom_jitter(aes(color = family), width = 0, height = 0.08, size = 2) + # endfold # fold: [Label the plot] labs(x = "Projected employment growth, 2025–35 (%)", y = NULL) + theme(legend.position = "none", axis.text.y = element_text(size = 12)) # endfold`

Student responses

1. `ai |> # fold: [Keep the highly exposed occupations] filter(exposure_percent >= 75) |> # endfold ggplot(aes(projected_growth, family)) + # fold: [Mark zero and draw one boxplot per family] geom_vline(xintercept = 0, color = "gray60") + geom_boxplot(outlier.shape = NA, width = 0.4) + # endfold # fold: [Show every occupation] geom_jitter(aes(color = family), width = 0, height = 0.08, size = 2) + # endfold # fold: [Label the plot] labs(x = "Projected employment growth, 2025–35 (%)", y = NULL) + theme(legend.position = "none", axis.text.y = element_text(size = 12)) # endfold`

Student responses

1. `ai |> # fold: [Keep the highly exposed occupations] filter(exposure_percent >= 75) |> # endfold ggplot(aes(projected_growth, family)) + # fold: [Mark zero and draw one boxplot per family] geom_vline(xintercept = 0, color = "gray60") + geom_boxplot(outlier.shape = NA, width = 0.4) + # endfold # fold: [Show every occupation] geom_jitter(aes(color = family), width = 0, height = 0.08, size = 2) + # endfold # fold: [Label the plot] labs(x = "Projected employment growth, 2025–35 (%)", y = NULL) + theme(legend.position = "none", axis.text.y = element_text(size = 12)) # endfold`

Student responses

1. `ai |> # fold: [Keep the highly exposed occupations] filter(exposure_percent >= 75) |> # endfold ggplot(aes(projected_growth, family)) + # fold: [Mark zero and draw one boxplot per family] geom_vline(xintercept = 0, color = "gray60") + geom_boxplot(outlier.shape = NA, width = 0.4) + # endfold # fold: [Show every occupation] geom_jitter(aes(color = family), width = 0, height = 0.08, size = 2) + # endfold # fold: [Label the plot] labs(x = "Projected employment growth, 2025–35 (%)", y = NULL) + theme(legend.position = "none", axis.text.y = element_text(size = 12)) # endfold`

Student responses

1. `ai |> # fold: [Keep the highly exposed occupations] filter(exposure_percent >= 75) |> # endfold ggplot(aes(projected_growth, family)) + # fold: [Mark zero and draw one boxplot per family] geom_vline(xintercept = 0, color = "gray60") + geom_boxplot(outlier.shape = NA, width = 0.4) + # endfold # fold: [Show every occupation] geom_jitter(aes(color = family), width = 0, height = 0.08, size = 2) + # endfold # fold: [Label the plot] labs(x = "Projected employment growth, 2025–35 (%)", y = NULL) + theme(legend.position = "none", axis.text.y = element_text(size = 12))` # Which kinds of highly exposed work are projected to grow?

Student responses

1. `ai |> # fold: [Keep the highly exposed occupations] filter(exposure_percent >= 75) |> # endfold ggplot(aes(projected_growth, family)) + # fold: [Mark zero and draw one boxplot per family] geom_vline(xintercept = 0, color = "gray60") + geom_boxplot(outlier.shape = NA, width = 0.4) + # endfold # fold: [Show every occupation] geom_jitter(aes(color = family), width = 0, height = 0.08, size = 2) + # endfold # fold: [Label the plot] labs(x = "Projected employment growth, 2025–35 (%)", y = NULL) + theme(legend.position = "none", axis.text.y = element_text(size = 12)) # Which kinds of highly exposed work are projected to grow?`
2. `ai |> filter(exposure >= 75) |> ggplot(aes(x = family, y = percent_change)) + geom_boxplot() + geom_jitter(width = 0.2) # Which kinds of highly exposed work are projected to grow?`
3. `ai |> filter(exposure_percent >= 75) |> ggplot() + geom_boxplot(aes(x = projected_growth, y = family) + geom_point() ) # Which kinds of highly exposed work are projected to grow?`

Student responses

1. `ai |> filter(exposure_percent >= 75) |> ggplot()+ geom_boxplot(aes(x = family, y = projected_growth)) + geom_point(aes(x=family, y=exposure_percent))`
2. `ai |> filter(exposure_percent >= 75) |> ggplot(aes(exposure_percent, family)) + geom_boxplot(outlier.shape = NA, width = 0.4) + geom_jitter(aes(color = family), width = 0, height = 0.08, size = 2) + labs(x = "AI exposure score", y = NULL) + theme(legend.position = "none", axis.text.y = element_text(size = 12))`
3. `ai |> filter(exposure_percent >= 75) |> ggplot(aes(projected_growth, family)) + geom_boxplot() + geom_point()`
4. `ai |> filter(exposure_percent >= 75) |> ggplot(aes(x = family, y = exposure_percent)) + geom_boxplot( fill = "#D8CEC4", color = "#7C746D" ) + geom_jitter( color = "#7D8B91", width = 0.15, height = 0 ) + labs( x = "Occupation Family", y = "AI Exposure Score" ) + theme_minimal()`

Student responses

1. `ai |> filter(exposure_percent >= 75) |> ggplot(aes(x = family, y = exposure_percent)) + geom_boxplot() + geom_jitter()+ labs( x = "Occupation Family", y = "AI Exposure (%)" ) # Which kinds of highly exposed work are projected to grow?`
2. `ai |> filter(exposure_percent >= 75) |> ggplot(aes(x = family, y = exposure_percent)) + geom_boxplot() + geom_jitter(width = 0.15, alpha = 0.6) # Which kinds of highly exposed work are projected to grow?`
3. `ai |> filter(exposure_percent >= 75) |> ggplot(aes(x = family, y = exposure_percent)) + geom_boxplot() + geom_jitter(width = 0.15, alpha = 0.7) + coord_flip()`
4. `ai |> filter(exposure_percent >= 75) |> ggplot(aes(x = family, y = exposure_percent)) + geom_boxplot() + geom_jitter(width = 0.2, alpha = 0.6)`
5. `ai |> filter(exposure_percent >= 75) |> ggplot(aes(x = family, y = exposure_percent)) + geom_boxplot() + geom_jitter(width = 0.2, alpha = 0.6) + labs(x = "Family", y = "Exposure percent") + theme(axis.text.x = element_text(angle = 45, hjust = 1))`

Student responses

1. `ai |> filter(exposure_percent >= 75) |> ggplot(aes(x = family, y = exposure_percent)) + geom_boxplot() + geom_jitter(width = 0.2) + labs(x = "Occupation family", y = "AI exposure score")`
2. `ai |> filter(exposure_percent >= 75) |> ggplot(aes(x = family, y = exposure_percent)) + geom_boxplot() + geom_point() # Which kinds of highly exposed work are projected to grow?`
3. `ai |> filter(exposure_percent >= 75) |> ggplot(aes(x = family, y = exposure_percent)) + geom_boxplot() + geom_point() # Which kinds of highly exposed work are projected to grow?`
4. `ai |> filter(exposure_percent >= 75) |> ggplot(aes(x = family, y = exposure_percent)) + geom_boxplot() + geom_point() # Which kinds of highly exposed work are projected to grow?`
5. `ai |> filter(exposure_percent >= 75) |> ggplot(aes(x = family, y = exposure_percent)) + geom_boxplot(outlier.shape = NA) + geom_jitter(width = 0.1, height = 0, alpha = .5) + labs(x = NULL, y = "AI exposure (%)") # Which kinds of highly exposed work are projected to grow?`

Student responses

1. `ai |> filter(exposure_percent >= 75) |> ggplot(aes(x = family, y = project_growth)) + geom_boxplot() + geom_point()` # Which kinds of highly exposed work are projected to grow?
2. `ai |> filter(exposure_percent >= 75) |> ggplot(aes(x = family, y = projected_growth)) + geom_boxplot(outlier.shape = NA) + geom_jitter(width = 0.15, height = 0) + labs( x = "Occupation family", y = "Projected employment growth, 2025–2035 (%)" )` # Which kinds of highly exposed work are projected to grow?
3. `ai |> filter(exposure_percent >= 75) |> ggplot(aes(x = family, y = projected_growth)) + geom_boxplot(outlier.shape = NA) + geom_point(position = position_jitter(width = 0.2, seed = 1), alpha = 0.6, color = "steelblue") + labs(x = "Occupation family", y = "Projected job growth", title = "Projected job growth for highly AI-exposed occupations") + theme_minimal() + coord_flip()`

Student responses

1. `ai |> filter(exposure_percent >= 75) |> ggplot(aes(x = projected_growth, y = family)) + geom_boxplot() + geom_point() # Which kinds of highly exposed work are projected to grow?`
2. `ai |> filter(exposure_score >= 75) |> ggplot(aes(x = family, y = projected_growth)) + geom_boxplot() + geom_jitter(width = 0.2) + labs( x = "Occupation family", y = "Projected growth" ) # Which kinds of highly exposed work are projected to grow?`
3. `ai |> filter(exposure_percent >= 75) |> ggplot() + geom_boxplot(aes(x = projected_growth, y = family)) # Which kinds of highly exposed work are projected to grow?`

Student responses

1. `ai |> filter(exposure_percent >= 75) |> # endfold ggplot(aes(projected_growth, family)) + # fold: [Mark zero and draw one boxplot per family]
 geom_vline(xintercept = 0, color = "gray60") + geom_boxplot(outlier.shape = NA, width = 0.4) + # endfold # fold: [Show every occupation]
 geom_jitter(aes(color = family), width = 0, height = 0.08, size = 2) + # endfold # fold: [Label the plot] labs(x = "Projected employment growth, 2025–35 (%)", y
 = NULL) + theme(legend.position = "none", axis.text.y = element_text(size = 12)) # Which kinds of highly exposed work are projected to grow?`
2. `ai |> filter(exposure_percent >= 75) |> arrange(desc(projected_growth)) |> slice_head(n=1) # Which kinds of highly exposed work are projected to grow?`
3. `ai |> filter(exposure_percent >= 75) |> ggplot() + geom_boxplot(aes(x = projected_growth, y = family)) + geom_boxplot() + geom_point() # Which kinds of
 highly exposed work are projected to grow?`

Student responses

1. `ai |> filter(exposure_percent >= 75) |> ggplot()+ geom_boxplot(aes(x = projected_growth, y = family)) + geom_point(aes(x = projected_growth, y = family))`
2. `ai |> filter(exposure_percent >= 75) |> ggplot(aes(projected_growth, family)) + # fold: [Mark zero and draw one boxplot per family] geom_vline(xintercept = 0, color = "gray60") + geom_boxplot(outlier.shape = NA, width = 0.4) + # endfold # fold: [Show every occupation] geom_jitter(aes(color = family), width = 0, height = 0.08, size = 2) + # endfold # fold: [Label the plot] labs(x = "Projected employment growth, 2025-35 (%)", y = NULL) + theme(legend.position = "none", axis.text.y = element_text(size = 12))`
3. `ai |> filter(exposure_percent >= 75) |> ggplot(aes(x = family, y = exposure_percent)) + geom_boxplot() + theme(axis.text.x = element_text(angle=45, vjust=.5, hjust=0.2)) + geom_point()` # Which kinds of highly exposed work are projected to grow?

Student responses

1. `ai |> filter(exposure_percent >= 75) |> ggplot(aes(x = projected_growth, y = family)) + geom_boxplot() + geom_jitter(width = 0.15, alpha = 0.6) # Which kinds of highly exposed work are projected to grow?`
2. `ai |> filter(exposure_percent >=75) |> ggplot(aes(x = family, y = exposure_percent)) + geom_boxplot() + geom_jitter(width = 0.2, alpha = 0.6) + labs(x = "Family", y = "Exposure percent") # Which kinds of highly exposed work are projected to grow?`
3. `ai |> filter(exposure_percent >= 75) |> ggplot(aes(x = family, y = exposure_percent)) + geom_boxplot() + geom_point()`
4. `ai |> filter(exposure_percent >= 75) |> ggplot() + geom_boxplot(aes(x=family, y=exposure_percent)) + geom_point(aes(x=family, y=exposure_percent)) ai |> filter(occupation == "Data scientists") # Which kinds of highly exposed work are projected to grow?`
5. `ai |> filter (exposure_percent >= 75) # Which kinds of highly exposed work are projected to grow?`

Student responses

1. `ai |> filter(exposure_percent >= 75) |> ggplot() + geom_boxplot(aes(x = projected_growth, y = family))` # Which kinds of highly exposed work are projected to grow?
2. `ai |> filter(exposure_percent >= 75) |> select(family) |> table()` # Which kinds of highly exposed work are projected to grow?
3. `ai |> filter(exposure_percent >= 75) |> ggplot() + geom_boxplot(aes(y = projected_growth, x = family)) |> geom_vline(xintercept = 0, color = "gray60") + geom_boxplot(outlier.shape = NA, width = 0.4) + # endfold # fold: [Show every occupation] geom_jitter(aes(color = family), width = 0, height = 0.08, size = 2) + # endfold # fold: [Label the plot] labs(x = "Projected employment growth, 2025–35 (%)", y = NULL) + theme(legend.position = "none", axis.text.y = element_text(size = 12))` # Which kinds of highly exposed work are projected to grow?
4. `ai |> filter(exposure_percent >= 75) |> ggplot() + geom_point(aes(x = exposure_percent, y = growth))` # Which kinds of highly exposed work are projected to grow?

Student responses

1. `ai |> filter(exposure_percent >= 75) |> ggplot(aes(x = projected_growth, y = family)) + geom_boxplot() + geom_point()` # Which kinds of highly exposed work are projected to grow?
2. `ai |> filter(exposure_percent >= 75) |> ggplot(aes(x = projected_growth, y = family)) + geom_boxplot() + geom_point()` # Which kinds of highly exposed work are projected to grow?
3. `ai |> filter(exposure_percent >= 75) |> arrange(projected_growth)` # Which kinds of highly exposed work are projected to grow?
4. `ai |> filter(exposure_percent >= 75) |> ggplot() + geom_boxplot( aes(x = projected_growth, family )` # Which kinds of highly exposed work are projected to grow?
5. `ai |> filter(exposure_percent >= 75) |> ggplot() + geom_point(aes(x = exposure_percent, y = projected_growth))` # Which kinds of highly exposed work are projected to grow?
6. `ai |> filter(exposure_percent >= 75) |> ggplot(aes(x = family, y = exposure_percent)) + geom_boxplot() + geom_jitter(width = 0.2)` # Which kinds of highly exposed work are projected to grow?

Student responses

1. `ai |> filter(exposure_percent >= 75) |> ggplot(aes(x = family, y = exposure_percent)) + geom_boxplot(outlier.shape = NA) + geom_jitter(width = 0.2, alpha = 0.4) + coord_flip() + labs(x = "Occupation family", y = "AI exposure (%)", title = "AI exposure by occupation family") # Which kinds of highly exposed work are projected to grow?`
2. `ai |> filter(exposure_percent >= 75) |> arrange(desc(projected_growth)) |> slice_head(n = 1) # Which kinds of highly exposed work are projected to grow?`
3. `ai |> filter(exposure_percent >= 75) |> ggplot(aes(x = family, y = exposure_percent)) + geom_boxplot() + geom_jitter(width = 0.15, alpha = 0.7) + coord_flip() # Which kinds of highly exposed work are projected to grow?`
4. `ai |> filter(exposure_percent >= 75) |> ggplot(aes(x = projected_growth, y = family)) + geom_boxplot() + geom_point() # Which kinds of highly exposed work are projected to grow?`
5. `ai |> filter(exposure_percent >= 75) |> ggplot() + geom_boxplot(aes(x=projected_growth, y = family)) # Which kinds of highly exposed work are projected to grow?`

Student responses

1. `ai |> filter(exposure_percent >= 75) |> ggplot(aes(x = family, y = exposure_percent)) + geom_boxplot() + geom_point(alpha = 0.4, position = "jitter") + labs(x = "Occupation family", y = "AI exposure score")`
2. `ai |> filter(exposure_percent >= 75)|> ggplot()+geom_boxplot(aes(x=projected_growth, y = family))+ geom_point() # Which kinds of highly exposed work are projected to grow?`
3. `ai |> filtered_data <- ai %>% filter(exposure_percent >= 75) |> ggplot(filtered_data, aes(x = occupation_family, y = exposure_score)) + geom_boxplot(outlier.shape = NA, fill = "lightblue", alpha = 0.5) + geom_jitter(width = 0.2, height = 0, alpha = 0.7) + labs(title = "Occupational Exposure Scores by Family", subtitle = "Occupations with exposure scores of 75 or higher", x = "Occupation Family", y = "Exposure Score")`

Student responses

1. `ai_filtered <- ai |> filter(exposure_percent >= 75) ggplot(ai_filtered, aes(x = family, y = exposure_percent)) + geom_boxplot() + geom_point(alpha = 0.5) + #`
Optional: adds points to see the distribution clearly `labs( title = "AI Exposure Score Distribution by Occupational Family (Scores >= 75)", x = "Occupational Family", y = "AI Exposure Percent" ) + theme(axis.text.x = element_text(angle = 45, hjust = 1)) # Rotates labels for readability`
2. `ai|> filter(exposure_percent >= 75)|> select(family)|> ggplot(aes(x = family, y = exposure_percent)) + geom_boxplot() + geom_jitter(width = 0.15, alpha = 0.7) + labs(x = "Family", y = "Exposure percent") + coord_flip() # Which kinds of highly exposed work are projected to grow?`
3. `ai|> filter(exposure_percent >= 75)|> arrange(desc(projected_growth))|> ggplot() + geom_boxplot(aes(y = projected_growth, x = family)) + geom_point() + labs(x = "Family", y = "Projected Growth", title = "Plot") # Which kinds of highly exposed work are projected to grow?`

Student responses

1. `ai|> filter(exposure_percent>=75)|> arrange(desc(projected_growth))` # Which kinds of highly exposed work are projected to grow?
2. `ai|>filter(exposure_percent >= 75)|> ggplot(aes(x=projected_growth,y=family))+ geom_boxplot()+ geom_point()` # Which kinds of highly exposed work are projected to grow?
3. `ai|>filter(exposure_percent >= 75)|> ggplot(aes(x = family, y = exposure_percent))+ geom_boxplot()+ geom_point()` ai # Which kinds of highly exposed work are projected to grow?
4. `high_exp <- ai |> filter(exposure_percent >= 75) ggplot(high_exp, aes(x = family, y = exposure_percent)) + geom_boxplot() + geom_jitter(width = 0.2, alpha = 0.6) + labs(x = "Occupation family", y = "Exposure score", title = "Highly exposed occupations by family") + theme(axis.text.x = element_text(angle = 45, hjust = 1))` # Which kinds of highly exposed work are projected to grow?

Student responses

1. `high_exposure <- ai |> filter(exposure_percent >= 75) ggplot(high_exposure, aes(x = family, y = exposure_percent)) + geom_boxplot() + geom_jitter(width = 0.2, alpha = 0.6) + labs(x = "Family", y = "Exposure score") + theme(axis.text.x = element_text(angle = 45, hjust = 1)) # Which kinds of highly exposed work are projected to grow?`
2. `high_jobs <- ai |> filter(exposure_percent >= 75) ggplot(high_jobs, aes(x = family, y = exposure_percent, fill = family)) + geom_boxplot(alpha = 0.7, outlier.shape = NA) + geom_jitter(width = 0.15, size = 2, alpha = 0.8, color = "black") + coord_flip() + labs(x = "Job family", y = "Exposure score", title = "Highly exposed occupations by job family") + theme_minimal() + theme(legend.position = "none", axis.text.y = element_text(size = 9)) # Which kinds of highly exposed work are projected to grow?`

Student responses

1. `library(dplyr) library(ggplot2) ai %>% filter(exposure_percent >= 75) %>% ggplot(aes(x = family, y = exposure_percent)) + geom_boxplot(outlier.shape = NA, fill = "lightblue") + geom_jitter(width = 0.15, height = 0, alpha = 0.6) + labs(x = "Occupation family", y = "AI exposure score") + theme_minimal() + theme(axis.text.x = element_text(angle = 45, hjust = 1))`
2. `nms <- names(ai) x_col <- nms[str_detect(nms, regex("expos", ignore_case = TRUE))][1] y_col <- nms[str_detect(nms, regex("growth|pct|percent|change", ignore_case = TRUE))][1] f_col <- nms[str_detect(nms, regex("family|group|category", ignore_case = TRUE))][1] ai |> filter(.data[[x_col]] >= 75) |> ggplot(aes(x = .data[[f_col]], y = .data[[y_col]])) + geom_boxplot(outlier.shape = NA) + geom_jitter(width = 0.2, alpha = 0.6) + coord_flip() + labs(x = "Occupational family", y = "Projected employment growth (%)") # Which kinds of highly exposed work are projected to grow?`

What would you tell a data science major who thinks high AI exposure means their field will shrink? Use the plot to support your answer.

0 anonymous responses

No responses were submitted.

If you could spend a year getting much better at one skill, which would you choose? Why do you think employers would value it?

82 anonymous responses

Working with AI

35

Learning to use or collaborate with artificial intelligence tools and systems.

Customer-facing and service skills

11

Improving abilities to interact with and support customers effectively.

Presentation skills

10

Improving ability to present ideas clearly and persuasively to audiences.

General communication

9

Enhancing verbal or interpersonal communication for clearer exchanges.

Response themes for question 9

Mathematics and problem solving**6**

Strengthening math knowledge to improve analytical thinking and problem-solving.

Networking, articulation, and profit**5**

Skills for promoting work, connecting with others, or generating profit that employers value.

Model building and technical modeling**3**

Developing skills to design and build predictive or analytical models.

Sports or nontechnical, AI-proof skill**3**

Developing athletic or other hands-on skills perceived as unlikely to be automated.

Student responses

1. AI
2. AI careers
3. AI knowldge with whatever profession you are going into
4. AI platforms and knowing how to effectively use AI.
5. AI stuff
6. ai use, employers want to hire new talent that can handle future technological advances to grow the company
7. Being able to effectively use agentic ai because it will make your work product exponentially faster
8. coding or enhanced working with ai, since ai use is on the uptick in the workplace
9. communication skills, like communicating with clients to explain what I did and why it is important for them

Student responses

1. data science, specifically the end of the pipeline that involves using machine learning to draw conclusions. We see that data science is expected to grow a lot.
2. fluency with ai
3. full stack development, even though AI can do it, trying to get AI to do just the right thing, and make it maintainable, makes it a good skill to have
4. getting better at Ai is important as it is getting incorporated into jobs
5. How to integrate AI too make myself more efficeint
6. I think the important idea to have is a flexibility and understanding of AI. This does not mean to be dependent on it but rather that you know how it works, how to use it properly, potentially how to build, and how to leverage it in your favor. This also goes hand in hand with understanding on maching learning also works
7. I think the most important skill to have right now in the workforce is working with AI.
8. i would choose ai modeling because i think employers value that you are able to use ai in an intelligent and efficient way

Student responses

1. I would choose critical thinking because employers value people who can solve problems, make good decisions, and use tools like AI effectively.
2. I would choose to improve my coding skills. Although AI can do a lot of the coding now, it's important for humans to understand the work that AI does so humans can interfere the process at any time.
3. I would choose to learn how to use AI
4. I would improve either my writing or my communication since they are both still important even if the AI can do a lot of the quantatative analysis.
5. I would spend my time in learning all about AI and how to use it to maximize efficiency because all employers are looking for ways to implement it to make more money for their company
6. increase humanity
7. Leveraging AI and being able to use resources well for problem solving/team work
8. Machine learning knowledge and critical reasoning, especially critical reasoning because AI has proven itself at being awful at it
9. Modeling, understanding code, and creative writing to help use AI to prompt the best possible outputs. Yes I do think employers would value it.

Student responses

1. People skills and communication is one of the most important skills, one that AI can not really reproduce. Besides that I honestly think that learning how to manipulate AI and master telling it what you want it to do can be a huge advantage that employers would value.
2. Probably getting good at using AI and being able to actually follow what AI is doing so that you will be able debug and make sure everything is working correctly so that no other mistakes happen
3. prompting. That would make me have better and more efficient cooperation with AI and agent tools.
4. the ability to utilize AI? because this will make my work more efficient
5. Using Ai in a way that optimizes productivity
6. Using AI to aid your work and also communicating your work
7. Using AI/automation to do mundane tasks. It's kind of taking over.
8. vibe coding skill and the ability to maximize ai usage and efficiency
9. working with ai

Student responses

1. Coding. With what I want to do, I wish I were able to know how to code everything and go from there.
2. Computer science? now a days life is made by cs
3. customer service
4. ETL pipelines and being able to work with and clean data that are stored in data warehouses, since AI models need huge amounts of data to work with in a cleaned manner
5. Finance understanding and ability to manage assets. Stock market is fluctuating a lot right now.
6. General technical literacy, including coding and software, informational graphics, and other things, because I plan on going into research
7. I think the most important skill in the work place is the ability to adapt and learn new things quick, with things constantly changing in the world these two skills get you ahead of everyone.
8. I would like to get better at paying attention, Ive gotten diagnosed with ADHD last year (womp womp)
9. I would like to master coding. Coding is so complex and difficult to me so it will be great if I can just know all kinds of coding

Student responses

1. I would spend a year getting better at proficiency in multiple programming languages and experience with common platforms
2. Probably would be a programming language, such as C++. I am proficient in Python, but C++ is a little lower level.

Student responses

1. A skill i would spend a year getting better at is communication and presentation skills. This is a skill an employer would value because it is something that cant be done using AI
2. good public speaking or communciation, cannot be replaced in any way with AI.
3. High AI exposure does not necessarily mean that a field will shrink. The plot shows that some highly AI-exposed occupations are still projected to experience positive employment growth. AI may complement workers and increase productivity rather than simply replace them.
4. I would choose data analysis. I think employers would value it because I could use data to answer questions and explain the results clearly.
5. I would choose programming skills
6. I would choose to improve my data analysis skills so I could turn raw information into useful insights. I think employers would value this because it would help me identify trends, solve problems, and support decisions with evidence. I would also practice explaining my findings clearly so others could use them.

Student responses

1. I would spend the year getting much better at communicating technical results to non-technical audiences. Employers value it because an analysis only creates value if decision-makers understand it and act on it; clear communication turns technical work into actual business decisions.
2. I would spend time practicing using AI, since employers are probably looking for more productive workers and ai can aid me in that. I'd also probably try to hone my presentation and communication skills since AI can't do that completely
3. I'd get better at public speaking, this does not go away as technology gets better, it actually becomes more important
4. Presentation skills

Student responses

1. Communication and creative thinking
2. communication love and peace
3. Communication of data. Being able to communicate to other people of multidisciplinary groups is something AI can not take over
4. Communication, hands down. One of the most important attributes of any system with people trying to work together is how they communicate issues, necessity of help, from general clarity to proper tone of voice.
5. i would get better at communication and soft skills because that is something ai cannot impact
6. I would say communication and the ability to work on a team because there's still that human aspect and EI that employers need to drive growth in a business
7. I would say social skill, but the market is more of occupied by existed monoploies, Or tech skills,, may be the hardware
8. Interpersonal skills, because it's so important to be able to communicate your findings as a data scientist and build networks
9. interviewing/people social skills,

Student responses

1. Get better at critical thinking, problem solving, collaboration, or creativity-- specific skills that will enhance your ability to use new tools
2. Lin alg intuition; would make my modeling skills of novel problems better.
3. math- helps with problem solving
4. Math. Knowing math is good and is applicable to lots of career like cs, data science, finance...
5. problem solving
6. Understanding of algorithms, we can see if AI did it right.

Student responses

1. Critical thinking, and using ai responsibly. Emploers will value as it is important skills nowadays
2. I would choose improving my background knowledge training for what ever job I'm going into.
3. Making profit; employers love profit
4. Networking
5. networking/articulation - doesn't matter if you're good at things if you can't talk about them in an accessible way

Student responses

1. creating models, models can be very effective at predicting results, which employers value
2. I would get better at model building and being able to create a story based on the numbers and share it in a way that makes sense to people who do not have the technical knowledge
3. machine learning, inference, model building - need someone to do the do communication

Student responses

1. Drawing conclusions from data and applying them to the real world.
2. I would pick soccer cuz then i could get paid for it and that is definately ai proof
3. I would spend a year getting better at designing experiments and using synthesis and materials characterization tools. This part of materials science is something AI currently cannot automate and should be valued by employers in my field.

Choose a skill. How would you investigate whether occupations requiring more of it tend to pay more? Describe a table or plot you would make.

82 anonymous responses

Plot skill importance versus pay

35

Suggests merging occupation wage and skill data and creating a plot (e.g., scatter or similar) of skill importance against pay to visualize association.

States a skill without method

11

Provides only the name of a skill without describing an investigation method or specific plot/table.

Compare pay of those with vs without skill

7

Proposes comparing pay for groups defined by presence or absence of the skill, often via bar graph.

Compare averages or groups by skill presence

6

Recommends comparing average pay between people or jobs with and without the skill or computing correlations/average differences rather than just plotting.

Response themes for question 10

Compute median pay by skill rating

6

Suggests combining skill ratings with wage data and summarizing pay (e.g., median) for each skill level or rating.

Fit a regression model

6

Proposes using linear regression to quantify the relationship between skill level and wages.

Use bar plot for categorical occupations

6

Notes that occupation is categorical and suggests a bar plot to display pay summaries by occupation.

Join tables then assess skill importance

5

Describes joining skill and occupation/wage tables and then assessing whether a particular skill is associated with higher pay, sometimes via correlation or comparison to other skills.

Student responses

1. A skill like communication can be used to measure the importance for every job by joining the three tables and making a plot
2. active listening i would plot the wages of the skills to its pay
3. active listening- I would make a scatterplot with how much active listening an occupation requires on the x axis and average pay on the y axis, with each point representing an occupation
4. Active Listening. So we receive the instructions more effective. I would make a scatter plot to see the trend.
5. Communication, plot of communication vs salary
6. complex problem solving, i would plot the skill level of the job required vs pay for each position to see a correlation, positive correlation shows requirement of it does pay more
7. Communication, i would investigate this by looking at wages, etc
8. everything matter in job, active learning, listening speaking whatever, cuz this is what this world is.
9. First, filter by occupations requiring that skill. Then, plot skill importance by wage with a point for each occupation.

Student responses

1. For a particular skill, make a scatter plot where x axis is the importance of that skill in a job and y axis is the pay. see if there are any relationships
2. For a skill like communication, I feel like that is a difficult thing to investiage whether requiring more of it tends to pay more, so I would plot skill improtance versus pay.
3. For the skill Installation, I would filter for the installation skill, and create a scatterplot with how important the skill is on the x axis and pay on the y axis.
4. graph of how much of a skill you have vs how much you make
5. I could make a plot of the most important skills for certain occupations then see if they correlate to higher pay
6. I select equipment installation. This is easy to identify in job requirements, and I would plot the relationship between jobs that require this and their pay.
7. I think i will study the active learning, I will at first the select the active_learning from the data set, And join it with table having the payment infor, I will make the plot graph, to show the correlation
8. I will make a scatter point comparing between the importance and pay

Student responses

1. I would choose communication. The plot would put communication rating on the x-axis and salary on the y axis.
2. I would choose data analysis and make a scatterplot with the required skill level on the x-axis and median annual pay on the y-axis, with one point per occupation. An upward trend would suggest that occupations requiring more data analysis tend to pay more, though it would not prove that the skill causes higher pay.
3. I would choose programming and join the skills and wages tables by occupation_code. I would make a scatterplot with programming importance on the x-axis and median annual pay on the y-axis, with one point per occupation. I would look for an upward trend to see whether jobs requiring more programming tend to pay more.
4. I would choose programming and make a scatterplot with programming importance on the x-axis and pay on the y-axis to see if jobs that require more programming tend to pay more.
5. I would join both tables by occupation code and make a plot of skill importance vs pay
6. I would make a graph of active learning vs income plot

Student responses

1. I would pick critical thinking. I would join the skills table to the wages table by occupation code, filter to the critical thinking rows, and make a scatterplot of median annual pay against the importance score with a trend line. If pay tends to rise as importance rises, that would suggest occupations requiring more critical thinking pay more.
2. I would plot a scatterplot with pay and importance
3. I would put skill level on the x axis of a scatterplot compared to pay. Each point in the plot would be a different occupation.
4. I would use a scatterplot of the average pay for occupations against the amount of the skill required
5. join wages and skills. Make a plot of importance and pay.
6. make a scatterplot with critical-thinking importance on the x-axis and annual pay on the y-axis, with one point for each occupation; and look for a positive relationship between skill importance and pay
7. plot the wage of that occupation against the importance of that particular skill for the job

Student responses

1. So this table would have a column for the number of times a particular skill show ups in a particular job posting for an occupation. That would be turned into a percentage, and another column would be the pay for that particular job posting or occupation (probably the former).
2. Sort by wages and create a coorleation between skills and jobs and see which jobs have high wages.
3. Taking each occupation as a observational unit, plot the median wage of the occupation against how important the skill is to the occupation.
4. $x = \text{skill name}$ $y = \text{income}$
5. You could make a plot of skill importance and pay for each of the skills

Student responses

1. 12 10 Management of Financial Resources 3.28 I would investigate this by looking at average income for those occupations in a table relative to other occupations that don't require it.
2. Communication skill. We could see how well they communicate their ideas to their pay and observe trends, patterns, and behaviors.
3. Complex problem solving and leaderships pay well as well as technical skills. The best way to check for relationship would be a scatterplot
4. Complex problem solving, through comparing its code with the amount of pay
5. Critical thinking.
6. Critical Thinking. I would filter the tables to be just critical thinking vs pay by joining the tables together
7. for active listening, i would see how much each occupation pays for each job
8. I think that active learning is quite important when it comes to the all occupations, since it showcase that your passion and your respect for others
9. mathematics. Investigate it by comparing the occupations with or without it on two different boxplots. plot skill importance vs pay

Student responses

1. Prompting. Filter out the occupations that require this skill and make a scatter plot of salary over importance score.
2. Sales. I would investigate by seeing how many companies are hiring sales people. I would make a ggplot

Student responses

1. A skill like hardware engineering would probably pay more. I'd make a bar graph comparing the pay of those with and without the skill.
2. Active learning. The would by testing regular to make sure the content was absorbed and understood. Use a scatterplot with critical-thinking importance on the x-axis and annual pay on the y-axis, one point per occupation. If occupations that require more critical thinking pay more, the cloud of points tilts upward from left to right. A flat or shapeless cloud would mean no real link. I could add a regression line to make the tilt easy to read.
3. I would analyze Judgement and Decision making by comparing the skill's importance across occupations, then I would tally the pay and compare it with another two skills. I would also compare past wages and current wages to evaluate if the skill gives a salary boost.
4. I would make a bar plot or a histogram to show the higher wages for people that have a certain skill. I would filter that skill and then plot how much those people with that skill get paid and compare

Student responses

1. I would use the occupations within each skill to calculate an average salary and then pair each average with the skill to see whether occupations requiring more of it tend to pay more. I would make a box plot dividing each plot by skill and on the y-axis put pay.
2. Let's say I chose communication. I would make a boolean column describing whether or not that skill was used in each profession, then make a series of boxplots of yes/no vs salary.
3. To investigate if jobs requiring high communication tend to pay more, I would first filter to only see communication and then sort my data to see what the top jobs are. I would then create another column that shows the rank of their salary.

Student responses

1. compare those who have skill and not and compare avg for both; boxplot to see distribution
2. compare what the pay is with an average person without the skill and average person with the skill (perhaps taking an average of the remaining skills weighted on what's most average)
3. look at occupation by skills, see year by year price increases based on more experience and increased skill, compare slopes across occupations/skills
4. Maybe if I chose active listening then I would group by it and see the average salary of those people. I would then maybe create a graph to see the comparison of that skill with other occupations
5. Sort by jobs with high skill in one of the skills and compare it with the average wage of that job
6. To investigate the skill of active listening, I would get a distribution of the occupations that require active listening compared to those that don't and calculate the correlation

Student responses

1. active learning, would get the jobs where that skill is the most important then average their salaries, compare it ith job salary mean of jobs that rank it as least important
2. find the skill and the rating combined with the skill, then find the median income associated with it
3. I would group the occupations by their score of how much they need the skill (rounding down) and then make a boxplot of score vs income
4. I would split the requirement level into three groups and make a box plot showing pay
5. Ill choose critical thinking, compare groups with higher scores for critical thinking against groups with less and see their pay difference
6. skill: communication. difficult to define but could group by communication?

Student responses

1. create regression for skills relative to pay
2. I would create a line graph and observe how groups of different levels of necessary occupation increase over time.
3. I would group the occupations and arrange the pay
4. I would join the two tables then run a regression to indicate the relationship between the importance of that skills and pay.
5. I would try to plot the time occupation requires and the pay of it, make as a scatter plot or column plot
6. make a linear regression

Student responses

1. Communication. Take mean salary for all the jobs and make a box plot.
2. i would filter to that skill and see the trend graph it out see the salary of epoepl haign that skill
3. I would have a table with a row for every occupation investigated, a column for whether it requires the skill or not, and the median pay for that occupation.
4. I would pick active listening, and you could investigate it by plotting all the jobs and seeing there value with active listening on the y axis and pay on the x
5. maybe a correlational plot to see the relationship between high need of a skill and where that pays high.
6. since skill importance is quantitative and pay is quantitative i would make a scatterplot

Student responses

1. I would compare the dependence of occupations on a skill against the wages earned by people in those occupations.
2. I would join wages and skills by occupation code, I would then choose a skill and see how correlated it is to payment numbers = scatter plot
3. I'll join the table skills with wages, using occupation code, than keep only my selected skill. The I would make a plot of wages agianst whether the occupation need this skill?
4. join the wages and skills tables then find the average wage for jobs where each skill is important and find the highest average.
5. Lets see the active listening skill. I guess we can check by joining this with a table of teh different occupations and also seeing if this skill compared to teh others has a larger pay than the other ones

Choose one skill and use `skills` and `wages` to make a scatterplot of its importance against annual pay, with one point per occupation. Label both axes and submit your plot.

54 anonymous responses

Other responses

54

Student responses

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.
- 7.
- 8.
- 9.

Student responses

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.
- 7.
- 8.
- 9.

Student responses

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.
- 7.
- 8.
- 9.

Student responses

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.
- 7.
- 8.
- 9.

Student responses

- 1.
- 2.
- 3.
- 4.
5.

```
# fold: [Join the tables and select critical thinking] skill_pay <- inner_join(skills, wages, by = "occupation_code") thinking <- skill_pay |> filter(skill == "Critical Thinking") # endfold thinking |> ggplot(aes(importance, pay)) + geom_point(alpha = .4) + # fold: [Label the axes] labs(x = "Critical-thinking importance (1-5)", y = "Annual pay ($1,000s)") # endfold
```
6. I added the title label
7. I chose mathematics as a skill with importance in x axis and pay in y axis
8. I don't know what's wrong with my plot,
9. It added a line.

Student responses

1. pay drifting upward as importance rises, the upward tilt is the sign that occupations needing more critical thinking tend to pay more
2. Persuasion plot!
3. programming as my skill
4. Scatterplot of median annual pay against the importance of critical thinking, one point per occupation. Pay tends to rise as the importance of critical thinking rises, which suggests occupations that require more critical thinking pay more.
5. `skills |> inner_join(wages, by = "occupation_code") |> filter(skill == 'Mathematics') |> select(occupation, pay, importance) |> ggplot() + geom_point(aes(x = importance, y = pay)) # Join the tables, choose a skill, and plot.`
6. `skills |> inner_join(wages, by = "occupation_code") |> filter(skill == "Complex Problem Solving") |> select(occupation, importance, pay) |> ggplot(aes(x = importance, y = pay)) + geom_point() + labs(x = "Importance", y = "Annual pay")`
7. `skills |> inner_join(wages, by = "occupation_code") |> filter(skill == "Mathematics") |> select(occupation, pay, importance) |> ggplot() + geom_point(aes(x = importance, y = pay))`
8. There is a noticeable outlier when importance is 3

Student responses

1. This is for equipment maintenance

Improve the vertical scale so the relationship among most occupations is easier to see. Keep all occupations visible, then submit your plot.

0 anonymous responses

No responses were submitted.

- Group the rows by skill. - Calculate the correlation between importance and pay within each skill. - Sort the correlations from highest to lowest and display the first eight skills.

75 anonymous responses

Other responses

75

Student responses

1. `# fold: [Calculate the correlation for each skill] skill_ranking <- skill_pay |> group_by(skill) |> summarize(correlation = cor(importance, pay)) # endfold
skill_ranking |> arrange(desc(correlation)) |> head(8)`
2. `# fold: [Calculate the correlation for each skill] skill_ranking <- skill_pay |> group_by(skill) |> summarize(correlation = cor(importance, pay)) # endfold
skill_ranking |> arrange(desc(correlation)) |> head(8)`
3. `skill_correlations <- skill_pay |> group_by(skill) |> summarize(correlation = cor(importance, pay)) skill_correlations |> arrange(desc(correlation)) |> head(8) #
Calculate a correlation for each skill, then rank the results.`
4. `skill_pay # Calculate a correlation for each skill, then rank the results.`
5. `skill_pay # Calculate a correlation for each skill, then rank the results.`
6. `skill_pay group_by(skill) |> summarise(correlation = cor(importance, pay, use = "complete.obs"), .groups = "drop") |> arrange(desc(correlation))|>
slice_head(n = 8) # Calculate a correlation for each skill, then rank the results.`

Student responses

1. `skill_pay skill_correlations <- skill_pay |> group_by(skill) |> summarize(correlation = cor(importance, annual_mean_wage, use = "complete.obs")) |> arrange(desc(correlation)) print(head(skill_correlations, 8))` # Calculate a correlation for each skill, then rank the results.
2. `skill_pay skill_pay |> group_by(skill) |> summarize(correlation = cor(importance, pay)) |> arrange(desc(correlation)) |> slice_head(n = 8)` # Calculate a correlation for each skill, then rank the results.
3. `skill_pay skills_pay %>% group_by(skill) %>% summarize(correlation = cor(importance, pay, use = "complete.obs")) %>% arrange(desc(correlation)) %>% slice_head(n = 8)` # Calculate a correlation for each skill, then rank the results.
4. `skill_pay |> group_by(skill) |> summarise(correlation = cor(importance, pay, use = "complete.obs")) |> arrange(desc(correlation)) |> head(8)` # Calculate a correlation for each skill, then rank the results.

Student responses

1. `skill_pay |> group_by(skill) |> summarise(correlation = cor(importance, pay)) |> arrange(desc(correlation)) |> slice_head(n=8)` # Calculate a correlation for each skill, then rank the results.
2. `skill_pay |> group_by(skill) |> summarize( correlation = cor(importance, pay) ) |> arrange(desc(correlation)) |> head(8)` # Calculate a correlation for each skill, then rank the results.
3. `skill_pay |> group_by(skill) |> summarize(corr = cor(importance, pay)) |> arrange(desc(corr))` # Calculate a correlation for each skill, then rank the results.
4. `skill_pay |> group_by(skill) |> summarize(corr = cor(importance, pay)) |> arrange(desc(corr)) |> slice_head(n = 8)` # Calculate a correlation for each skill, then rank the results.
5. `skill_pay |> group_by(skill) |> summarize(correlation = cor(importance, pay, use = "complete.obs")) |> arrange(desc(correlation)) |> head(8)` # Calculate a correlation for each skill, then rank the results.

Student responses

1. `skill_pay |> group_by(skill) |> summarize(correlation = cor(importance, pay)) |> arrange(desc(correlation))` # Calculate a correlation for each skill, then rank the results.
2. `skill_pay |> group_by(skill) |> summarize(correlation = cor(importance, pay)) |> arrange(desc(correlation)) |> head(3)`
3. `skill_pay |> group_by(skill) |> summarize(correlation = cor(importance, pay)) |> arrange(desc(correlation)) |> head(8)` # Calculate a correlation for each skill, then rank the results.
4. `skill_pay |> group_by(skill) |> summarize(correlation = cor(importance, pay)) |> arrange(desc(correlation)) |> slice_head(n = 8)`
5. `skill_pay |> group_by(skill) |> summarize(correlation = cor(importance, pay)) |> arrange(desc(correlation)) |> slice_head(n = 8)` # Calculate a correlation for each skill, then rank the results.
6. `skill_pay |> group_by(skill) |> summarize(correlation = cor(importance, pay)) |> arrange(desc(correlation)) |> slice_head(n = 8)` # Calculate a correlation for each skill, then rank the results.

Student responses

1. `skill_pay |> group_by(skill) |> summarize(correlation = cor(importance, pay)) |> arrange(desc(correlation)) |> slice_head(n=8)` # Calculate a correlation for each skill, then rank the results.
2. `skill_pay |> group_by(skill) |> summarize(r = cor(importance, pay, use = "complete.obs")) |> arrange(desc(r)) |> head(8)` # Calculate a correlation for each skill, then rank the results.
3. `skill_pay |> group_by(skill) |> summarize(corr = cor(importance, pay)) |> arrange(desc(corr))` # Calculate a correlation for each skill, then rank the results.
4. `skill_pay |> group_by(skill) |> mutate(corr = cor(importance, pay)) |> select(skill, importance, pay, corr) |> arrange(desc(corr))` # Calculate a correlation for each skill, then rank the results.
5. `skill_pay |> group_by(skill) |> summarize( corr = cor(importance, pay) ) |> arrange(desc(corr))` # Calculate a correlation for each skill, then rank the results.
6. `skill_pay |> group_by(skill) |> summarize(corr = cor(importance, pay)) arrange(desc(corr)) |>` # Calculate a correlation for each skill, then rank the results.

Student responses

1. `skill_pay |> group_by(skill) |> summarize(corr = cor(importance, pay)) |> arrange(desc(corr))` # Calculate a correlation for each skill, then rank the results.
2. `skill_pay |> group_by(skill) |> summarize(corr = cor(importance, pay)) |> arrange(desc(corr)) |> slice_head(n = 20)` # Calculate a correlation for each skill, then rank the results.
3. `skill_pay |> group_by(skill) |> summarize(corr = corr(importance, pay)) |> arrange(desc(corr))` # Calculate a correlation for each skill, then rank the results.
4. `skill_pay |> group_by(skill) |> summarize(correlation = cor(importance, pay)) |> arrange(desc(correlation)) |> head(8)` # Calculate a correlation for each skill, then rank the results.
5. `skill_pay |> group_by(skill) |> summarize(correlation = cor(importance, pay)) |> arrange(desc(correlation)) |> head(8)` # Calculate a correlation for each skill, then rank the results.
6. `skill_pay |> group_by(skill) |> summarize(correlation = cor(importance, pay)) |> arrange(desc(correlation)) |> slice_head(n = 8)` # Calculate a correlation for each skill, then rank the results.

Student responses

1. `skill_pay |> group_by(skill) |> summarize(correlation = cor(importance, pay)) |> arrange(desc(correlation)) |> slice_head(n = 8)` # Calculate a correlation for each skill, then rank the results.
2. `skill_pay |> group_by(skill) |> summarize(correlation = cor(importance, pay)) |> arrange(desc(correlation)) |> slice_head(n = 8)` # Calculate a correlation for each skill, then rank the results.
3. `skill_pay |> group_by(skill) |> summarize(correlation = cor(importance, pay)) |> arrange(desc(correlation)) |> slice_head(n=8)` # Calculate a correlation for each skill, then rank the results.
4. `skill_pay |> group_by(skill)|> summarize(correlation = cor(importance, pay)) |> arrange(desc(correlation)) |> slice_head(n=8)` # Calculate a correlation for each skill, then rank the results.
5. `skill_pay |> group_by(skill) |> summarize(corr = cor(importance, pay)) |> arrange(desc(corr)) |> head(n = 8)` # Calculate a correlation for each skill, then rank the results.

Student responses

1. `skill_pay |> group_by(skill) |> summarize(correlation = cor(importance, pay)) |> arrange(desc(correlation)) |> slice_head(n=8)` # Calculate a correlation for each skill, then rank the results.
2. `skill_pay |> group_by(skill) |> summarize(correlation = cor(importance, pay)) |> arrange(desc(correlation))` # Calculate a correlation for each skill, then rank the results.
3. `skill_pay |> group_by(skill) |> summarize( corr = cor(importance, pay) ) |> arrange(desc(corr))` # Calculate a correlation for each skill, then rank the results.
4. `skill_pay |> group_by(skill) |> summarize(corr = cor(importance, pay)) |> arrange(desc(corr))` # Calculate a correlation for each skill, then rank the results.
5. `skill_pay |> group_by(skill) |> summarize(corr = cor(importance, pay)) |> arrange(desc(corr))` # Calculate a correlation for each skill, then rank the results.
6. `skill_pay |> group_by(skill) |> summarize(corr = cor(importance, pay)) |> arrange(desc(corr)) |>` # Calculate a correlation for each skill, then rank the results.

Student responses

1. `skill_pay |> group_by(skill) |> summarize(correlation = cor(importance, pay)) |> arrange(desc(correlation)) |> slice_head(n = 8)` # Calculate a correlation for each skill, then rank the results.
2. `skill_pay |> group_by(skill) |> summarize(corr = cor(importance, pay)) |> arrange(desc(corr)) |> head(8)` # Calculate a correlation for each skill, then rank the results.
3. `skill_pay |> group_by(skill) |> summarize(correlation = cor(importance, pay)) |> arrange(desc(correlation)) |> slice_head(n=8)` # Calculate a correlation for each skill, then rank the results.
4. `skill_pay |> group_by(skill) |> summarize( corr = cor(importance, pay) ) |> arrange(desc(corr))` # Calculate a correlation for each skill, then rank the results.
5. `skill_pay |> group_by(skill) |> summarize(corr = cor(importance, pay)) |> arrange(desc(corr))`
6. `skill_pay |> group_by(skill) |> summarize(corr = cor(importance, pay)) |> arrange(desc(corr))` # Calculate a correlation for each skill, then rank the results.

Student responses

1. `skill_pay |> group_by(skill) |> summarize(corr = cor(importance, pay)) |> arrange(desc(corr))` # Calculate a correlation for each skill, then rank the results.
2. `skill_pay |> group_by(skill) |> summarize(corr = cor(importance, pay)) |> arrange(desc(corr))` # Calculate a correlation for each skill, then rank the results.
3. `skill_pay |> group_by(skill) |> summarize(corr = cor(importance, pay)) |> arrange(desc(corr)) |> slice_head(n = 8)`
4. `skill_pay |> group_by(skill) |> summarize(corr = cor(importance, pay)) |> arrange(desc(corr)) |>` # Calculate a correlation for each skill, then rank the results.
5. `skill_pay |> group_by(skill) |> summarize(corr = cor(importance,pay)) |> arrange(desc(corr))` # Calculate a correlation for each skill, then rank the results.
6. `skill_pay |> group_by(skill) |> summarize(corr = cor(importance,pay)) |> arrange(desc(corr))` # Calculate a correlation for each skill, then rank the results.
7. `skill_pay |> group_by(skill) |> summarize(correlation = cor(importance, pay)) |> arrange(desc(correlation))` # Calculate a correlation for each skill, then rank the results.

Student responses

1. `skill_pay |> group_by(skill) |> summarize(correlation = cor(importance, pay)) |> arrange(desc(correlation)) |> slice_head( n = 8)` # Calculate a correlation for each skill, then rank the results.
2. `skill_pay |> group_by(skill) |> summarize(corr = cor(importance,pay)) |> arrange(desc(corr)) |> slice(8)` # Calculate a correlation for each skill, then rank the results.
3. `skill_pay |> group_by(skill) |> summarise(correlation = cor(importance, pay, use = "complete.obs")) %>% arrange(desc(correlation)) %>% slice_head(n = 8)`
4. `skill_pay |> group_by(skill) |> summarize(c = cor(importance, pay)) |> arrange(c)` # Calculate a correlation for each skill, then rank the results.
5. `skill_pay |> group_by(skill) |> summarize(cor = cor(importance, pay)) |> arrange(desc(cor)) |> head(8)`
6. `skill_pay |> group_by(skill) |> summarize(correlation = cor(importance, pay)) |> arrange(desc(correlation))` # Calculate a correlation for each skill, then rank the results.

Student responses

1. `skill_pay |> group_by(skill) |> summarize(correlation = cor(importance, pay)) |> arrange(desc(correlation))` # Calculate a correlation for each skill, then rank the results.
2. `skill_pay |> group_by(skill) |> summarize(correlation = cor(pay, importance)) |> arrange(desc(correlation))` # Calculate a correlation for each skill, then rank the results.
3. `skill_pay |> group_by(skill)|> summarize(corr = cor(importance,pay))|> arrange(desc(corr))` # Calculate a correlation for each skill, then rank the results.
4. `skill_pay|> group_by(skill)|> summarize(corr = cor(importance, pay))|> arrange(desc(corr))` # Calculate a correlation for each skill, then rank the results.
5. `skill_pay|> group_by(skill) |> summarize(corr = cor(importance, pay))` # Calculate a correlation for each skill, then rank the results.
6. `skill_pay|> group_by(skill)|> summarize(corr = cor(importance, pay))|> arrange(desc(corr))` # Calculate a correlation for each skill, then rank the results.
7. `skill_pay|> glimpse()` # Calculate a correlation for each skill, then rank the results.

Student responses

1. `skill_pay|>group_by(skill)|>summarize(corr = cor(importance,pay))|>arrange(desc(corr))` # Calculate a correlation for each skill, then rank the results.
2. `skill_pay|>group_by(skill)|>summarize(corr = cor(impoartance, pay))|>arrange(desc(corr))` # Calculate ad correlation for each skill, then rank the results.
3. `skill_pay|>group_by(skill)|>summarize(correlation = cor(importance, pay)) arrange(desc(correlation)) |> slice_head(n = 8)` # Calculate a correlation for each skill, then rank the results.
4. `skill_pay|>group_by(skill)|>summarize(correlation = cor(importance, pay))|>arrange(desc(correlation))` # Calculate a correlation for each skill, then rank the results.
5. `skill_ranking <- skill_pay |> group_by(skill) |> summarize(correlation = cor(importance, pay)) skill_ranking |> arrange(desc(correlation)) |> head(8)` # Calculate a correlation for each skill, then rank the results.

Find the skill you chose at the beginning. Where does it appear in the ranking? Is this what you expected?

78 anonymous responses

Brief numeric rank report Student gives only a short numeric rank without commentary.	15
Alignment with expectations Student reports that the skill's rank matches what they expected.	13
Skill ranked lower than expected Student notes the chosen skill appears lower in the ranking than they anticipated.	13
Expresses mismatch and frustration Student emphasizes a strong perceived lack of correlation and self-critical surprise that the skill did not match expectations.	8

Response themes for question 14

Reports early/high rank as expected**8**

Student indicates the skill appears near the top of the ranking and that matches their expectation.

Names a specific skill with score**7**

Student provides the name of a specific skill and a numeric score but gives no further commentary.

Skill not found / different skill type**7**

Student cannot find their chosen skill on the list and notes they were thinking of a different (typically more technical) skill set.

Surprised technical skill ranked low**7**

Student is surprised that a technical skill like programming is not ranked among the top skills or high-paying ones.

Student responses

1. 3 Complex Problem Solving 0.625, yes what I expected because its valuable no matter the field
2. 4th, I know that Active learning is important
3. Active Learning appears at 4 in the list, which is what I expected
4. critical thinking: 0.625
5. Complex Problem Solving was third in the ranking, which is to be expected, as someone who can do this at a high level is more likely to be making a lot of money
6. Complex problem solving was top 3 in terms of correlation between importance and pay. This was expected, since the scatter plot showed a positive trend between the two.
7. Critical thinking is second and that is about where I expected it to be
8. high up
9. I chose complex problem solving which is ranked 3rd which is around where I expected

Student responses

1. It appears secopnd in the ranking which is what i expected
2. it is #4 yes
3. its 3rd very correlated and yes very expected
4. the first skill i chose is active listening, it shows up at the top
5. the fourth one
6. third, yes i thought it would be decently high

Student responses

1. #2, yes the scatterplot showed a strong positive correlation, matched the pattern seen in the plot
2. Critical thinking appears second on the list which makes sense.
3. Critical Thinking is rank 2, I expected it to be important
4. Critical thinking is ranked pretty high and it is what I expected.
5. Critical Thinking ranks near the top of the skills most positively associated with higher pay. This is consistent with my expectation because critical thinking is especially important in many complex, high-skilled occupations.
6. I chose critical thinking. It turns out the top 10 skills in correlation. Therefore, it aligns with my expectation.
7. It appears In the middle/top as communication is a pretty crucial aspect that determines pay
8. It is second with correlation 0.625 which is where I expected it.
9. It was 11th on the ranking, and it is about where I expected.

Student responses

1. It was a little different from I expected but overall the same
2. Its second on the list and yes it was expected tp be a the very top somewhere
3. The skill I chose was Judgement and Decision Making, which was 1st in the ranking. It was what I had expected because considering how much information is out there, having good judgement and decision making skills is increasingly important to not be replaced by AI.
4. Yes, complex problem solving ranked 2nd which is kinda what I expected because its an important skill to have no matter the occupation

Student responses

1. Communication is not in the top 10 which surprised me
2. Equipment maintenance was definitely way lower than I expected. I suppose this is because it can be treated as its own job in positions that only require labor.
3. I chose writing which appears about where i expected it in the higher coorelated skill with pay.
4. It appears lower in the ranking because most soft skills show up at the top which makes sense because those are skills that would be required for all jobs
5. It did not appear where I expected in the ranking, I thought it would be higher.
6. It is lower than I thought it would be
7. It is not in the top 10
8. it ranked lower than expected
9. it's in fourth place, i expected that since it's a wanted skill

Student responses

1. Its at 4 shich is whag i expected
2. My chosen skill is programming and it didn't appear on the list. That's not really what I expected
3. Persuasion was around 25 in the rankings with a correlation at about .3, which is roughly what I would expect.
4. Ranked lower than expected, it was 25th.

Student responses

1. 14 Active Listening 0.422, a bit off my expectation
2. every skill
3. I investigated critical thinking. It ranks 2nd out of the 35 skills for correlation between importance and pay. That is about what I expected, since critical thinking is central to high-skill professional and managerial work, though I was a little surprised it came in second rather than first.
4. I picked equipment maintenance, that sucks its last
5. I would say it is relatively correlated. Paying attention to things is necessary in order to do critical thinking
6. incredibly NOT correlated, im stupid
7. It has a correlation of 0.42, which is about what I thought, it is important but also a very basic skill
8. My skill is very similar to judgement which is found at the top of the ranking

Student responses

1. I picked writing which is pretty highly correlated with high pay which was surprising to me.
2. I was actually expecting certain specialized skill most important,, HOWEVER<<< it is not,, it is still those general thinking skills....
3. I would choose critical thinking and join the skills and wages tables by occupation code. I would make a scatterplot with critical thinking importance on the x-axis and annual pay on the y-axis, with one point per occupation. An upward trend would suggest that occupations requiring more critical thinking tend to pay more, though it would not prove that the skill causes higher pay.
4. It actually lists at 4th. Yes the active learning is pretty important no matter what the context is
5. It was pretty high up, which is what I expected
6. It's near the beginning, which I expected
7. Math appears near the top of the rankings, which is what I expected
8. Math does not even appear in th rank. I thought it was going to be higher thant hi

Student responses

1. Active Listening 0.422 I cannot figure out how to find the rank too D:
2. Active listening is correlated at 0.422 with pay. It's number 10 on the list. I expected it to be there
3. I don't remember the skill i chose in the beginning, but I would expect correlation for critical thinking to be somewhat higher? .6 is ok but not that strong of a correlation for what I would think.
4. I had active learning and it is towards the highest on the list of the highest correlation, which I guess goes to show that this skill is pretty important in getting a job. The skill was active learning
5. It has a correlation of 0.383, which is about what I expected given the graph.
6. It ranks 11. That makes sense because AI is taking away quantitative needs.
7. writing is close, it's in the top ten which is what I expected

Student responses

1. appears in middle, mathematics .4
2. Communication is in the middle of the pack, and I thought it would be higher.
3. critical thinking and problem solving are some of the highest ranked skills, which is what I expected
4. I can't find it on the list, because I was thinking about more technical skills than soft skills
5. I chose active learning and found out that it is on the top of the list. I am not surprised at this
6. It doesn't appear. I believe my initial logic on how this would appear is different, because correlation seems to be a stronger metric for this case.
7. it was not on the list. it is not what i expected. maybe it was not in the survey

Student responses

1. 8 th? science? Maybe cs is in science. Not what I expected
2. Critical thinking is very correlated, second most of all the skills included. I honestly thought it would not be as high as it is, even though I did think it was important.
3. I chose active listening; it's surprisingly high in the ranking at 0.462 correlation. This is not what I expected, but it is very cool.
4. I chose communication in the beginning
5. I picked science which as a correlation of 0.527 and appears 8th on the list. This is what I expected because physical science often requires a large skillset within itself.
6. it is what I expected because its suppose to be popular
7. Programming did not make the top eight skills, which surprised me because I expected it to be more strongly connected with higher pay.

- Fit pay using importance as the predictor, and save the model. - Display the intercept and slope. - Plot the observations and add the fitted regression line.

79 anonymous responses

Other responses

79

Student responses

1. `# fold: [Fit the critical-thinking pay model] pay_model <- lm(pay ~ importance, data = thinking) ggplot(thinking, aes(x=importance, y = pay)) + geom_point() + geom_smooth(method = "lm") # endfold coef(pay_model)`
2. `df <- lm(pay ~ importance, data = thinking) |> coef() ggplot(thinking, aes(x = importance, y = pay)) + geom_point(color = "pink") + geom_smooth(color = "purple", method = "lm") # Fit the model, inspect its coefficients, and draw the fitted line.`
3. `fit <- lm(pay ~ importance, data = thinking) |> coef() ggplot(thinking) + geom_point( aes(x = importance, y = pay)) + geom_smooth(method = "lm") # Fit the model, inspect its coefficients, and draw the fitted line.`
4. `fit <- lm(thinking$pay ~ thinking$importance) fit |> coef() ggplot(thinking) + geom_point(aes(x=importance, y=pay)) + geom_smooth(method = "lm") # Fit the model, inspect its coefficients, and draw the fitted line.`

Student responses

1. `library(ggplot2) # Fit and save the model model <- lm(pay ~ importance, data = thinking) # Display the intercept and slope coef(model) # Plot observations and the fitted regression line ggplot(thinking, aes(x = importance, y = pay)) + geom_point() + geom_abline(intercept = coef(model)[1], slope = coef(model)[2], color = "blue") + labs(x = "Importance of critical thinking", y = "Annual pay ($)") + theme_minimal() # Fit the model, inspect its coefficients, and draw the fitted line.`
2. `library(tidyverse) # 1. Fit and save the model pay_model <- lm(pay ~ importance, data = thinking) # 2. Intercept and slope coef(pay_model) # 3. Observations plus the fitted line ggplot(thinking, aes(x = importance, y = pay)) + geom_point(alpha = 0.4) + geom_abline(intercept = coef(pay_model)[1], slope = coef(pay_model)[2], color = "steelblue", linewidth = 1)`

Student responses

1. `lm <- lm(pay ~ importance, data = thinking) lm$coefficients ggplot(thinking, aes(x = importance, y = pay)) + geom_point() + geom_smooth(method = "lm")` # Fit the model, inspect its coefficients, and draw the fitted line.
2. `lm(pay ~ importance, data = thinking) coef() ggplot(thinking, aes(x = importance, y = pay)) + geom_point() + geom_smooth(method = "lm")` # Fit the model, inspect its coefficients, and draw the fitted line.
3. `lm(pay ~ importance, data = thinking) -> my_lm coef(my_lm) ggplot(thinking, aes(x=importance,y=pay)) + geom_point() + geom_smooth(method = "lm")` # Fit the model, inspect its coefficients, and draw the fitted line.
4. `lm(pay ~ importance, data = thinking) |> coef() ggplot(thinking, aes(x = importance, y = pay)) + geom_point() + geom_smooth(method = "lm")` # Fit the model, inspect its coefficients, and draw the fitted line.
5. `lm(pay ~ importance, data = thinking) |> coef() ggplot ggplot(thinking, aes(x = importance, y = pay)) + geom_point() + geom_smooth(method = "lm")` # Fit the model, inspect its coefficients, and draw the fitted line.

Student responses

1. `lm(pay ~ importance, data = thinking) |> coef()` `ggplot(thinking, aes(x = importance, y = pay)) + geom_point() + geom_smooth(method = "lm")` # Fit the model, inspect its coefficients, and draw the fitted line.
2. `lm(pay ~ importance, data = thinking) |> coef()` `ggplot(thinking, aes(x = importance, y = pay)) + geom_point() + geom_smooth(method = "lm")` # Fit the model, inspect its coefficients, and draw the fitted line.
3. `lm(pay ~ importance, data = thinking) |> coef()` `|> ggplot(thinking, aes(x = importance, y = pay)) + geom_point()` # Fit the model, inspect its coefficients, and draw the fitted line.
4. `lm(pay ~ importance, data = thinking) |> coef()` `ggplot(thinking, aes(x = importance, y = pay)) + geom_point() + geom_smooth(method = "lm")` # Fit the model, inspect its coefficients, and draw the fitted line.
5. `lm(pay ~ importance, data = thinking) |> coef()` `ggplot(thinking, aes(x = importance, y = pay)) + geom_point() + geom_smooth(method = "lm")` # Fit the model, inspect its coefficients, and draw the fitted line.

Student responses

1. `lm(pay ~ importance, data = thinking) |> coef()` `ggplot(thinking, aes(x = importance, y = pay)) + geom_point() + geom_smooth(method = "lm")` # Fit the model, inspect its coefficients, and draw the fitted line.
2. `lm(pay ~ importance, data = thinking) |> coef()` `ggplot(thinking, aes(x = importance, y = pay)) + geom_point() + geom_smooth(method = "lm")` # Fit the model, inspect its coefficients, and draw the fitted line.
3. `lm(pay ~ importance, data = thinking) |> coef()` `ggplot(thinking, aes(x = importance, y = pay)) + geom_point() + geom_smooth(method='lm')` # Fit the model, inspect its coefficients, and draw the fitted line.
4. `lm(pay ~ importance, data = thinking) |> coef()` `ggplot(thinking, aes(x = importance, y = pay)) + geom_point() + geom_smooth(method = "lm")` # Fit the model, inspect its coefficients, and draw the fitted line.
5. `lm(pay ~ importance, data = thinking) |> coef()` `ggplot(thinking, aes(x = importance, y= pay)) + geom_point() + geom_smooth(method = "lm")`

Student responses

1. `lm(pay ~ importance, data = thinking) |> coef()` `ggplot(thinking, aes(x=importance, y=pay)) + geom_point() + geom_smooth(method = "lm")` # Fit the model, inspect its coefficients, and draw the fitted line.
2. `lm(pay ~ importance, data = thinking) |> coef()` `ggplot(thinking, aes(importance, pay)) + geom_point() + geom_smooth(method = "lm")`
3. `lm(pay ~ importance, data = thinking) |> coef()` `lm(pay ~ importance, data = thinking) |> intercept()` `ggplot(thinking, aes(x = importance, y = pay)) + geom_point() + geom_smooth(method = "lm")` # Fit the model, inspect its coefficients, and draw the fitted line.
4. `lm(pay ~ importance, data = thinking) |> coef()` `ggplot(thinking, aes(x = importance, y = pay)) + geom_point() + geom_smooth(method = "lm")` # Fit the model, inspect its coefficients, and draw the fitted line.
5. `lm(pay ~ importance, data = thinking) |> coef()` `ggplot(thinking, aes(x = importance, y = pay))+ geom_point() + geom_smooth(method = "lm")` # Fit the model, inspect its coefficients, and draw the fitted line.

Student responses

1. `lm(pay ~ importance, data = thinking) |> coef() summary() ggplot() + geom_point(data = thinking, aes(x = importance, y = pay)) + geom_smooth(method = "lm")`
Fit the model, inspect its coefficients, and draw the fitted line.
2. `lm(pay ~ importance, data = thinking) |> coef() ggplot(thinking, aes(x = importance, y = pay)) + geom_point() + geom_smooth(method = "lm")` # Fit the model, inspect its coefficients, and draw the fitted line.
3. `lm(pay ~ importance, data = thinking) |> coef() ggplot(thinking, aes(x=important,y=pay)) + geom_point() + geom_smooth(method = "lm")` # Fit the model, inspect its coefficients, and draw the fitted line.
4. `lm(pay ~ importance, data = thinking) |> coef() ggplot(thinking, aes(x = importance, y = pay)) + geom_point() + geom_smooth(method = "lm")` # Fit the model, inspect its coefficients, and draw the fitted line.
5. `lm(pay ~ importance, data = thinking) |> coef() ggplot(thinking, aes(x=importance, y=pay)) + geom_point() + geom_smooth(method = "lm")` # Fit the model, inspect its coefficients, and draw the fitted line.

Student responses

1. `lm(pay ~ importance, data = thinking) |> coef()` # Fit the model, inspect its coefficients, and draw the fitted line.
2. `lm(pay ~ importance, data = thinking) |> coef()` `ggplot(thinking, aes(x= importance, y = pay)) + geom_point() + geom_smooth(method = "lm")` # Fit the model, inspect its coefficients, and draw the fitted line.
3. `lm(pay ~ importance, data = thinking) |> coef()` `ggplot(thinking, aes(x=importance, y = pay)) + geom_point() + geom_smooth(method = "lm")` # Fit the model, inspect its coefficients, and draw the fitted line.
4. `lm(pay ~ importance, data = thinking)|> coef()` `ggplot(thinking, aes(x = importance, y= pay))+ geom_point() + geom_smooth(method = "lm")` # Fit the model, inspect its coefficients, and draw the fitted line.
5. `lm(pay ~ importance, data = thinking)|> coef()` `ggplot(thinking, aes(x=importance, y = pay)) + geom_point() + geom_smooth(method = "lm")` # Fit the model, inspect its coefficients, and draw the fitted line.

Student responses

1. `lm(pay ~ importance, data = thinking)|> coef() ggplot(thinking,aes(importance,pay))+ geom_point() +geom_smooth(method = lm) # Fit the model, inspect its coefficients, and draw the fitted line.`
2. `lm(pay ~ importance, data = thinking)|> coef() ggplot(thinking, aes(x = importance, y = "pay"))+ geom_point() + geom_smooth(method = "lm") # Fit the model, inspect its coefficients, and draw the fitted line.`
3. `lm(pay ~ importance, data = thinking)|> coef() ggplot(thinking,aes(x = importance, y = pay))+ geom_point()+ geom_smooth(method = "lm") # Fit the model, inspect its coefficients, and draw the fitted line.`
4. `lm(pay~importance, data = thinking) |> coef() ggplot(thinking, aes(x = importance, y = pay)) + geom_point() + geom_smooth(method = "lm") # Fit the model, inspect its coefficients, and draw the fitted line.`
5. `lm(pay~importance, data=thinging) |> coef() ggplot(thinking, aes(x=importance, y = pay)) + geom_point() + geom_smooth(method="lm") # Fit the model, inspect its coefficients, and draw the fitted line.`

Student responses

1. `lm(pay~importance,data = thinking) |> coef() ggplot(thinking,aes(x = importance, y = pay))+geom_point()+ geom_smooth(method = "lm")`
2. `m1 <-lm(pay ~ importance, thinking) coef(m1) ggplot(thinking, aes(x = importance, y = pay)) + geom_point() + geom_smooth(method = "lm")` # Fit the model, inspect its coefficients, and draw the fitted line.
3. `meow <- lm(pay ~ importance, data = thinking) meow |> coef() |> print() ggplot(thinking, aes(x=importance, y=pay)) + geom_point() + geom_smooth(method = "lm")` # Fit the model, inspect its coefficients, and draw the fitted line.
4. `new_pay <- lm(pay ~ importance, data = thinking) |> coef(new_pay) ggplot(thinking, aes( x = importance, y = pay )) + geom_point() + geom_smooth(method = "lm")` # Fit the model, inspect its coefficients, and draw the fitted line.
5. `pay_correlation<- lm(pay~importance, data = thinking)|> coef() ggplot(thinking, aes(x=importance, y=pay))+ geom_point()+geom_smooth(method = "lm")` # Fit the model, inspect its coefficients, and draw the fitted line.

Student responses

1. `pay_model <- lm(pay ~ importance, data = jobs) coef(pay_model) plot(jobs$importance, jobs$pay, xlab = "importance", ylab = "pay", main = "Pay vs Importance") abline(pay_model, col = "blue", lwd = 2)`
2. `pay_model <- lm(pay ~ importance, data = thinking) coef(pay_model) ggplot(thinking, aes(importance, pay)) + geom_point(alpha = .4) + geom_abline(intercept = coef(pay_model)[1], slope = coef(pay_model)[2], color = "red", linewidth = 1) + labs(x = "Critical-thinking importance (1–5)", y = "Annual pay ($1,000)") # Fit the model, inspect its coefficients, and draw the fitted line.`
3. `pay_model <- lm(pay ~ importance, data = thinking) coef(pay_model) ggplot(thinking, aes(x = importance, y = pay)) + geom_point() + geom_smooth(method = "lm", se = FALSE) # Fit the model, inspect its coefficients, and draw the fitted line.`

Student responses

1. `pay_model <- lm(pay ~ importance, data = thinking) coef(pay_model) ggplot(thinking, aes(x = importance, y = pay)) + geom_point() + geom_smooth(method = "lm") # Fit the model, inspect its coefficients, and draw the fitted line.`
2. `pay_model <- lm(pay ~ importance, data = thinking) coef(pay_model) plot(thinking$importance, thinking$pay, xlab = "importance", ylab = "pay")
abline(pay_model, col = "blue", lwd = 2) # Fit the model, inspect its coefficients, and draw the fitted line.`
3. `pay_model <- lm(pay ~ importance, data = thinking) coef(pay_model) thinking |> ggplot(aes(importance, pay)) + geom_point(alpha = .4) +
geom_smooth(method = "lm", se = FALSE) + labs(x = "Critical-thinking importance (1-5)", y = "Annual pay ($1,000s)") # Fit the model, inspect its coefficients,
and draw the fitted line.`
4. `pay_model <- lm(pay ~ importance, data = thinking) coef(pay_model) thinking |> ggplot(aes(importance, pay)) + geom_point(alpha = .4) +
geom_smooth(method = "lm", se = FALSE) + labs(x = "Critical-thinking importance (1-5)", y = "Annual pay")`

Student responses

1. `pay_model <- lm(pay ~ importance, data = thinking) coef(pay_model) thinking |> ggplot(aes(x = importance, y = pay)) + geom_point() + geom_smooth(method = "lm", se = FALSE)`
2. `pay_model <- lm(pay ~ importance, data = thinking) coef(pay_model) thinking |> ggplot(aes(x = importance, y = pay)) + geom_point(alpha = .4) + geom_smooth(method = "lm", se = FALSE)`
3. `pay_model <- lm(pay ~ importance, data = thinking) coef(pay_model) thinking |> ggplot(aes(x = importance, y = pay)) + geom_point(alpha = 0.4, color = "steelblue") + geom_smooth(method = "lm", formula = y ~ x, se = FALSE) + labs(x = "Critical-thinking importance (1-5)", y = "Annual pay ($1,000s)") # Fit the model, inspect its coefficients, and draw the fitted line.`
4. `pay_model <- lm(pay ~ importance, data = thinking) coef(pay_model) thinking |> ggplot(aes(x = importance, y = pay)) + geom_point(alpha = 0.4) + geom_smooth(method = "lm", se = FALSE) + labs(x = "Critical-thinking importance (1-5)", y = "Annual pay ($1,000s)") + theme_minimal()`

Student responses

1. `pay_model <- lm(pay ~ importance, data = thinking) coef(pay_model) ggplot(thinking, aes(x = importance, y = pay)) + geom_point() + geom_smooth(method = "lm", se = FALSE)`
2. `pay_model <- lm(pay ~ importance, data = thinking) coef(pay_model) ggplot(thinking, aes(x = importance, y = pay)) + geom_point() + geom_smooth(method = "lm")`
3. `pay_model <- lm(pay ~ importance, data = thinking) coef(pay_model) ggplot(thinking, aes(x = importance, y = pay)) + geom_point() + geom_smooth(method = "lm")` # Fit the model, inspect its coefficients, and draw the fitted line.
4. `pay_model <- lm(pay ~ importance, data = thinking) coef(pay_model) ggplot(thinking, aes(x = importance, y = pay)) + geom_point() + geom_smooth(method = "lm")`
5. `pay_model <- lm(pay ~ importance, data = thinking) coef(pay_model) plot(thinking$importance, thinking$pay, xlab = "importance", ylab = "pay") abline(pay_model, col = "blue", lwd = 2)` # Fit the model, inspect its coefficients, and draw the fitted line.

Student responses

1. `pay_model <- lm(pay ~ importance, data = thinking) ggplot(thinking, aes(x = importance, y = pay)) + geom_point() + geom_smooth(method = "lm")` # Fit the model, inspect its coefficients, and draw the fitted line.
2. `pay_model <- lm(pay ~ importance, data = thinking) |> coef() ggplot(thinking, aes(x = importance, y = pay)) + geom_point() + geom_smooth(method = "lm")` # Fit the model, inspect its coefficients, and draw the fitted line.
3. `pay_model <- lm(pay ~ importance, data = thinking); coef(pay_model); thinking |> ggplot(aes(importance, pay)) + geom_point(alpha = 0.4) + geom_smooth(method = "lm", se = FALSE)`
4. `pay_model <- lm(pay ~ importance, data = your_data) coef(pay_model) plot(your_data$importance, your_data$pay, xlab = "importance", ylab = "pay") abline(pay_model, col = "blue", lwd = 2)` The intercept is `coef(pay_model)[1]` The slope is `coef(pay_model)[2]`

Student responses

1. `pay_model <- thinking |> lm(pay ~ importance, data = skills) coef(pay_model) plot(skills$importance, skills$pay) abline(pay_model, col = "blue", lwd = 2)` If you want to label the axes: `plot(skills$importance, skills$pay, xlab = "Importance of Use thinking", ylab = "Pay") abline(pay_model, col = "blue", lwd = 2)` # Fit the model, inspect its coefficients, and draw the fitted line.
2. `pay_model <- thinking |> lm(pay ~ importance) coef(pay_model) thinking |> ggplot(aes(x = importance, y = pay)) + geom_point() + geom_smooth(method = "lm", se = FALSE)` # Fit the model, inspect its coefficients, and draw the fitted line.
3. `pay_model = lm(pay ~ importance, data = thinking) |> coef(pay_model) ggplot(thinking, aes(x = importance, y = pay)) + geom_point() + geom_smooth(method = "lm")` # Fit the model, inspect its coefficients, and draw the fitted line.
4. `sharpmodel <- lm(pay ~ importance, data = thinking) coef(sharpmodel) ggplot(thinking, aes(x = importance, y = pay)) + geom_point() + geom_smooth(method = "lm", se = FALSE)`

Student responses

1. thinking # Fit the model, inspect its coefficients, and draw the fitted line.
2. thinking pay_model <- lm(pay ~ importance, data = thinking) coef(pay_model) plot(thinking\$importance, thinking\$pay) abline(pay_model, col = "blue", lwd = 2) # Fit the model, inspect its coefficients, and draw the fitted line.
3. thinking lm(pay ~ importance, data = thinking) |> coef() ggplot(thinking, aes(x = importance, y = pay)) + geom_point() + geom_smooth(method = "lm") # Fit the model, inspect its coefficients, and draw the fitted line.
4. thinking lm(pay~importance, data=thinking)|> coef() ggplot(thinking,aes(x=importance,y=pay))+ geom_point()+geom_smooth(method="lm") # Fit the model, inspect its coefficients, and draw the fitted line.

Student responses

1. `thinking pay_model <- lm(annual_mean_wage ~ importance, data = thinking) coef(pay_model) ggplot(thinking, aes(x = importance, y = annual_mean_wage)) + geom_point(alpha = 0.5, color = "steelblue") + geom_abline(intercept = coef(pay_model)[1], slope = coef(pay_model)[2], color = "darkorange", size = 1) + labs(title = "Annual Pay vs. Thinking Skill Importance", x = "Importance Score", y = "Annual Mean Wage") + theme_minimal() # Fit the model, inspect its coefficients, and draw the fitted line.`
2. `thinking pay_model <- lm(pay ~ importance, data = thinking) coef(pay_model) thinking |> ggplot(aes(x = importance, y = pay)) + geom_point() + geom_smooth(method = "lm", se = FALSE) + labs( x = "Importance", y = "Pay" ) # Fit the model, inspect its coefficients, and draw the fitted line.`

Student responses

1. `thinking pay_model <- lm(pay ~ importance, data = thinking) coef(pay_model) ggplot(thinking, aes(x = importance, y = pay)) + geom_point(alpha = 0.4) + geom_smooth(method = "lm", se = FALSE) + labs(x = "Critical-thinking importance (1-5)", y = "Annual pay (USD)") # Fit the model, inspect its coefficients, and draw the fitted line.`
2. `thinking pay_model <- lm(pay ~ importance, data = your_data) coef(pay_model) plot(your_data$importance, your_data$pay, xlab = "importance", ylab = "pay") abline(pay_model, col = "blue", lwd = 2) # Fit the model, inspect its coefficients, and draw the fitted line.`
3. `thinking_model <- lm(pay ~ importance, data = thinking) coef(thinking_model) thinking |> ggplot(aes(x = importance, y = pay)) + geom_point() + geom_smooth(method = "lm", se = FALSE) + labs(x = "Importance of Critical Thinking", y = "Annual Pay") # Fit the model, inspect its coefficients, and draw the fitted line.`

Two occupations have critical-thinking importance scores of 3 and 4. What difference in annual pay does this line predict, in dollars?

0 anonymous responses

No responses were submitted.

If we refit the regression without the highest-paid occupation, what do you expect to happen to the slope and R-squared? Explain your prediction.

0 anonymous responses

No responses were submitted.

- Save all but the highest-paid occupation as `thinking_without_top`. - Regress pay on importance using those occupations; save the model as `pay_without_top`. - Display the new model's coefficients and R-squared.

76 anonymous responses

Other responses

76

Student responses

1. `## fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(____) # endfold`
`thinking # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = ____)`
`# endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)$r.squared # endfold`
2. `# fold: [Exclude the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < max(pay)) # endfold`
`# fold: [Refit the regression] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold`
`coef(pay_without_top) summary(pay_without_top)$r.squared`
3. `# fold: [Exclude the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < max(pay)) # endfold`
`# fold: [Refit the regression] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold`
`coef(pay_without_top) summary(pay_without_top)$r.squared # endfold`

Student responses

1. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < 500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold
2. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay != max(pay, na.rm = TRUE)) pay_without_top <- lm(pay ~ importance, data = thinking_without_top) coef(pay_without_top) summary(pay_without_top)\$r.squared
3. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < max(pay)) pay_without_top <- lm(pay ~ importance, data = thinking_without_top) coef(pay_without_top) summary(pay_without_top)\$r.squared

Student responses

1. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < 500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold
2. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < 500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold

Student responses

1. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < 500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_data) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold
2. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < 500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, # fold: [Keep all but the highest-paid occupation] thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold

Student responses

1. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < 500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold
2. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < 500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold

Student responses

1. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < 500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold
2. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay<500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold
3. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(____) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = ____) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold

Student responses

1. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(____) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = ____) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold
2. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(____) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold

Student responses

1. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(____) # endfold thinking_without_top <- thinking |> filter(annual_mean_wage != max(annual_mean_wage, na.rm = TRUE)) pay_without_top <- lm(annual_mean_wage ~ importance, data = thinking_without_top) coef(pay_without_top) summary(pay_without_top)\$r.squared # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = ____) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold

Student responses

1. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(____) # endfold thinking_without_top <- thinking |> filter(pay < max(pay)) pay_without_top <- lm(pay ~ importance, data = thinking_without_top) coef(pay_without_top) summary(pay_without_top)\$r.squared # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = ____) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold
2. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(occupation != "Surgeon") # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold

Student responses

1. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(occupation != "Surgeons") # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold
2. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay != max(pay)) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold

Student responses

1. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < 245) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold
2. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < 500) # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold
3. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < 500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold

Student responses

1. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < 500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold
2. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < 500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold summary(pay_without_top) # endfold
3. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < 500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)

Student responses

1. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < 500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold
2. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < 500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold

Student responses

1. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < 500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold
2. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < 500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold

Student responses

1. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < 500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold
2. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < 500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold

Student responses

1. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < 500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold
2. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < 500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold

Student responses

1. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < 500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold
2. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < 500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold

Student responses

1. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < 500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold
2. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < 500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold

Student responses

1. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < 500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold
2. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < 500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold

Student responses

1. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < 500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold
2. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < 500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold

Student responses

1. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < 500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold
2. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < 500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold

Student responses

1. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < 500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold
2. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < 500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold

Student responses

1. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < 500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold
2. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < 500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold

Student responses

1. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < 500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold
2. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < 500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold

Student responses

1. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < 500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold
2. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < 500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold

Student responses

1. `# fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < 500000) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)$r.squared # endfold`
2. `# fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < 500000) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)$r.squared # endfold`

Student responses

1. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < max(pay, na.rm = TRUE)) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold
2. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < max(pay)) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = _____) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold

Student responses

1. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < max(pay)) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold
2. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < max(pay)) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold

Student responses

1. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < max(pay)) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold
2. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay < max(pay)) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data =thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold

Student responses

1. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay <= 400) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold
2. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay > 100) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold

Student responses

1. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay<500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold
2. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay<500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold

Student responses

1. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay<500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold
2. # fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay<500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)\$r.squared # endfold

Student responses

1. `# fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay<500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_without_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)$r.squared # endfold`
2. `# fold: [Keep all but the highest-paid occupation] thinking_without_top <- thinking |> filter(pay<500) # endfold # fold: [Fit the regression to the remaining occupations] pay_without_top <- lm(pay ~ importance, data = thinking_wthout_top) # endfold # fold: [Inspect the coefficients and R-squared] coef(pay_without_top) summary(pay_without_top)$r.squared # endfold`
3. `thinking_without_top <- thinking |> dplyr::filter(pay < max(pay, na.rm = TRUE)) pay_without_top <- lm(pay ~ importance, data = thinking_without_top) coef(pay_without_top) summary(pay_without_top)$r.squared # endfold`

Student responses

1. `thinking_without_top <- thinking |> filter(pay < max(pay, na.rm = TRUE)) pay_without_top <- lm(pay ~ importance, data = thinking_without_top)
coef(pay_without_top) summary(pay_without_top)$r.squared # endfold`
2. `thinking_without_top <- thinking |> filter(pay < max(pay, na.rm = TRUE)) pay_without_top <- lm(pay ~ importance, data = thinking_without_top)
coef(pay_without_top) summary(pay_without_top)$r.squared # endfold`
3. `thinking_without_top <- thinking |> filter(pay < max(pay)) pay_without_top <- lm(pay ~ importance, data = thinking_without_top) coef(pay_without_top)
summary(pay_without_top)$r.squared # endfold`
4. `thinking_without_top <- thinking |> filter(pay < max(pay)) pay_without_top <- lm(pay ~ importance, data = thinking_without_top) coef(pay_without_top)
summary(pay_without_top)$r.squared`
5. `thinking_without_top <- thinking |> filter(pay < max(pay)); pay_without_top <- lm(pay ~ importance, data = thinking_without_top); coef(pay_without_top);
summary(pay_without_top)$r.squared`

What do you expect the slope to be if we regress pay on **Complex Problem Solving** importance instead of **Critical Thinking** importance? Why?

78 anonymous responses

Expect similar slope

35

Student expects the regression slope to be similar to the original, indicating little or no change when swapping predictors.

Expect steeper/stronger slope

14

Student expects a stronger or steeper positive relationship between Complex Problem Solving importance and pay than for Critical Thinking.

Expect a lower slope

12

Student predicts the slope will decrease, implying a weaker positive association with pay for Complex Problem Solving importance.

Unspecified different slope

7

Student states the slope will differ but gives no direction or justification.

Response themes for question 19

Other responses	6
Responses the model could not place reliably.	
Other responses	4

Student responses

1. I expect the slope to be close to the critical-thinking slope, roughly \$50,000 per importance point, maybe a touch lower. The reason is that the two skills move together across occupations. Jobs that lean hard on critical thinking are almost always the same jobs that lean hard on complex problem solving, they are both the analytical, decision-heavy skills.
2. I expect the slope to be positive and similar in size to the Critical Thinking slope.
3. I expect the slope to be positive and similar to the critical-thinking slope because occupations that require more complex problem solving are also likely to be higher-skilled and higher-paying.
4. I expect the slope to be similar because these skills are similar.
5. I expect the slope to be similar, since these are similar skills and thus may display similar wage patterns.
6. I expect the slope to be similar.
7. I suspect we will see a very similar model, since the two skills seem pretty similar
8. I think it was pretty similar since they had similar correlation values from my memory (but if they have different correlation values, it might be a bit different)

Student responses

1. I think it will be relatively similar since they were ranked similarly in the importance table
2. I think it would be about the same, because the ranking of this skill is similar to critical thinking, and the jobs that it is applicable to should be very similar.
3. I think it would be similar because Complex Problem Solving and Critical Thinking are two very similar things.
4. I think it would be the same because they are pretty similar
5. I think that the slope will be similar since the skills are not far apart and similar themselves.
6. I think that we will get similar results but it will be maybe a little less in the coefficient. Complex problems require you to critically think
7. i think the corr would be similar because these two things go hand in hand
8. I think the results will be relatively similar since it was also ranked highly on the correlation rankings
9. I think the slope and r^2 will increase because i think complex problem solving is a more valuable skill.

Student responses

1. I think the slope will be positive and relatively steep. Kinda like the critical thinking plot
2. I think the slope will be somewhat similar or more steep because it showed a similar correlation but I'd also expect this skill to be way less useful in lower paying jobs.
3. I think the slope will stay very similar since those two skills are so interconnected.
4. I think the slope would be about the same since the two are pretty similar
5. I think the slope would change but not much as these skills both help job in similar amount.
6. I thinkk the slope would be similar since they are of similar type
7. I would expect a similar slope becuase those skills are pretty equal in value to employers
8. I would expect the slope to be positive and similar to critical thinking because jobs that require more complex problem solving also tend to pay more.
9. If we regress pay on Complex Problem Solving, I expect the slope to be similar

Student responses

1. it will probably have a similar regression model since they are similar skills.
2. It would be around the same as they had similarly strong correlations, but it will be slightly different depending on the actual values in the data
3. similar
4. Similar
5. Similar results? Because critical thinking is pretty similar to complex problem solving where both involves a lot of skills
6. slope would probably be similar slope because these two skills would go hand in hand and they are required for most occupations and they were ranked pretty similarly
7. Something similar will happen because as you naturally become very good at a skill firms are willing to pay you more
8. The slope may be a little flatter but it should stay around the same
9. The slope will probably be pretty similar because the skills are associated and similar to each other.

Student responses

1. Higher since it had more importance
2. I expect the slope to be positive and about the same as for Critical Thinking, maybe a little steeper, because jobs that need complex problem solving are usually the same higher paying jobs that need critical thinking
3. I expect the slope to be deeper because I remember complex problem solving has higher pay compared
4. I think it would decrease because people at the lower end of hierarchies get the annoying, hard to solve problems/jobs
5. I think that the slope will increase
6. I would expect the slope to be higher because Complex Problem Solving was a slightly higher ranked skill
7. increase, since its ranking is higher than critical thinking one
8. It might increase. I feel like pay bumps are attributed more to if you can more effectively solve problems. That's the more direct thing than just critical thinking.
9. it will be higher? i think because complex problem solving as more of an impact

Student responses

1. It would be more because complex problem solving seems more applicable to situations that are more tangible and critical thinking is more of an abstract skill
2. slope would increase since critical thinking is decreasing?
3. stronger
4. The slope will stay positive but it will be steeper
5. There will be a higher r squared value and a greater regression line, because Problem Solving was the highest correlated with pay when we tested earlier.

Student responses

1. decrease
2. I expect it to be lower since its less correlated and lower ranked.
3. I expect the slope coefficient to be lower
4. I think the slope will be lower
5. I think the slope would be a little less steep, as we already saw that critical thinking had a higher correlation with pay from our previous data searches.
6. It will be slightly lower because complex problem solving was ranked less important than critical thinking
7. slope will be less cus coefficent less
8. Slope will decrease because complex problem solving is of less importance than critical thinking.
9. smaller,, less extreme value

Student responses

1. the slope will change, become less steep
2. The slope would be smaller because there is less of an importance of complex problem solving than critical thinking and people are not paid as much based on this trait.
3. We'd see a lower slope, but we cannot predict the R^2 value. We know that the slope would be lower as we have already seen critical thinking is one of the top skills (if not the top item) to have.

Student responses

1. Correlations will be different and possibly the relationship between those two variables
2. I feel like they are fairly similar in terms of relationship to pay but I would say a subtle change, not sure if positive or negative.
3. it will be different
4. The correlation will change, and because the correlation was greater in the table before, the slope will be higher than before
5. we don't know the exact different slope that it could have, but there is a different slope I believe
6. We would have a different graph because our residual values and line of best fit have changed with the variable change.
7. would change slightly, since it's likely these two are correlated as well

Student responses

1. I would expect a positive slope, similar in size to the critical thinking slope, roughly \$45-50 thousand per one-point increase in importance. Complex problem solving is also one of the skills most strongly associated with higher pay, so pay should rise with its importance too. It might be a touch smaller than the critical thinking slope since critical thinking ranked slightly higher in the correlation ranking.
2. I would expect the slope to be positive, since complex problem solving is highly correlated with pay.
3. slide away
4. The slope will will lean much toard ccomplex problem solving
5. They would be similar.
6. This depends. If the two variables, complex problem solving and critical thinking are correlated, then the answer will probably be somewhat the same, but if they are not, I think we would see an opposite effect

Student responses

1. I expect that complex problem solving may have the more strong correlation with the critical thinking skill
2. I think complex problem solving is a different kind of ability, so itll be different.
3. Slop should be positive
4. We may recieve less growth trend pattern, as the complex can be helped with ai

If we use both critical thinking and problem solving to predict pay, do you expect us to explain substantially more variation? Why?

0 anonymous responses

No responses were submitted.

- Reshape two_skills into both_skills, with one occupation per row and separate thinking and problem_solving columns containing importance scores. - Fit pay using both skill scores, and save the model as both_model. - Display the model's coefficients and R-squared.

0 anonymous responses

No responses were submitted.

Both slopes fell substantially, while R-squared rose from 39% to 41%. How would you explain that? What does this suggest about interpreting the original \$51,000 slope as the value of critical thinking?

0 anonymous responses

No responses were submitted.

- Find the three occupations with the largest positive residuals. - Find the three with the most negative residuals. - For each group, display occupation, importance, pay, and residual.

62 anonymous responses

Other responses

62

Student responses

1. # 3 occupations paying more than predicted thinking |> arrange(desc(residual)) |> slice_head(n = 3) |> select(occupation, importance, pay, residual) # 3 occupations paying less than predicted thinking |> arrange(residual) |> slice_head(n = 3) |> select(occupation, importance, pay, residual)
2. # Largest positive residuals: paid more than predicted thinking |> arrange(desc(residual)) |> select(occupation, importance, pay, residual) |> head(3) # Most negative residuals: paid less than predicted thinking |> arrange(residual) |> select(occupation, importance, pay, residual) |> head(3)
3. # Three largest positive residuals thinking |> arrange(desc(residual)) |> select(occupation, importance, pay, residual) |> slice_head(n = 3) # Three most negative residuals thinking |> arrange(residual) |> select(occupation, importance, pay, residual) |> slice_head(n = 3) # Find occupations above and below the model's predictions.

Student responses

1. `largest_pos <- thinking |> arrange(desc(residual)) |> slice_head(n=3) |> select(occupation, importance, pay, residual)`
`largest_neg <- thinking |> arrange(residual) |> slice_head(n=3) |> select(occupation, importance, pay, residual)` # Find occupations above and below the model's predictions.
2. `largest_positive <- thinking %>% arrange(desc(residual)) %>% slice_head(n = 3) %>% select(occupation, importance, pay, residual)`
`most_negative <- thinking %>% arrange(residual) %>% slice_head(n = 3) %>% select(occupation, importance, pay, residual)`
3. `largest_positive <- thinking |> mutate(residual = pay - predict(pay_model)) |> arrange(desc(residual)) |> slice_head(n = 3) |> select(occupation, importance, pay, residual)`
`most_negative <- thinking |> mutate(residual = pay - predict(pay_model)) |> arrange(residual) |> slice_head(n = 3) |> select(occupation, importance, pay, residual)`
`largest_positive` `most_negative` # Find occupations above and below the model's predictions.

Student responses

1. `model <- lm(pay ~ importance, data = thinking) jobs_resid <- thinking |> mutate(residual = resid(model)) largest_positive <- jobs_resid |> select(occupation, importance, pay, residual) |> arrange(desc(residual)) |> slice_head(n = 3) most_negative <- jobs_resid |> select(occupation, importance, pay, residual) |> arrange(residual) |> slice_head(n = 3) largest_positive most_negative # Find occupations above and below the model's predictions.`
2. `new_thinking <- thinking |> mutate(residual = pay - predict(pay_model)) |> top_n(3, residual) new_thinking # Find occupations above and below the model's predictions.`
3. `pay_model <- lm(pay ~ importance, data = thinking); thinking |> mutate(residual = residuals(pay_model)) |> arrange(desc(residual)) |> head(3) |> select(occupation, importance, pay, residual); thinking |> mutate(residual = residuals(pay_model)) |> arrange(residual) |> head(3) |> select(occupation, importance, pay, residual)`

Student responses

1. `pos_thinking <- thinking |> mutate(residual = pay - predict(pay_model)) |> arrange(desc(residual)) |> slice_head(n = 3)`
`neg_thinking <- thinking |> mutate(residual = pay - predict(pay_model)) |> arrange(residual) |> slice_head(n = 3)`
`pos_thinking neg_thinking` # Find occupations above and below the model's predictions.
2. `resid_data <- thinking |> mutate(residual = resid(pay_model)) |> select(occupation, importance, pay, residual)`
`bind_rows( top_pos = slice_max(resid_data, residual, n = 3), top_neg = slice_min(resid_data, residual, n = 3), .id = "group" )` # Find occupations above and below the model's predictions.
3. `skills |> inner_join(wages, by = "occupation_code") |> filter(skill == "Critical Thinking") |> mutate(residual = pay - predict(pay_model)) |> top_n(3, residual)`
4. `skills |> inner_join(wages, by = "occupation_code") |> filter(skill == "Critical Thinking") |> mutate(residual = pay - predict(pay_model)) |> top_n(3, residual)` # Find occupations above and below the model's predictions.

Student responses

1. `skills |> inner_join(wages, by = "occupation_code") |> filter(skill == "Critical Thinking") |> mutate(residual = pay - predict(pay_model)) |> top_n(3, residual) #`
Find occupations above and below the model's predictions.
2. `skills |> inner_join(wages, by = "occupation_code")|> filter(skill == "Critical Thinking")|> mutate(residual = pay - predict(pay_model))|> top_n(3, residual) #`
Find occupations above and below the model's predictions.
3. `skills |> inner_join(wages, by = "occupation_code") |> filter(skill == "Critical Thinking") |> mutate(residual = pay - predict(pay_model)) |> top_n(3, residual) #`
Find occupations above and below the model's predictions.
4. `skills |> inner_join(wages, by = "occupation_code") |> filter(skill == "Critical Thinking") #` Find occupations above and below the model's predictions.
5. `skills|> inner_join(wages, by = "occupation_code")|> filter(skill == "Critical Thinking")|> mutate (residual = pay - predict(pay_model))|> top_n (3, residual) #`
Find occupations above and below the model's predictions.

Student responses

1. `skills|> inner_join(wages,by="occupation_code")|> filter(skill=="Critical thinking")` # Find occupations above and below the model's predictions.
2. `thinking` # Find occupations above and below the model's predictions.
3. `thinking` # Find occupations above and below the model's predictions.
4. `thinking` # Find occupations above and below the model's predictions.
5. `thinking model <- lm(importance ~ pay, data = thinking) jobs_resid <- thinking |> mutate(residual = resid(model)) largest_positive <- jobs_resid |> select(occupation, importance, pay, residual) |> slice_max(order_by = residual, n = 3) most_negative <- jobs_resid |> select(occupation, importance, pay, residual) |> slice_min(order_by = residual, n = 3) largest_positive most_negative` # Find occupations above and below the model's predictions.

Student responses

1. `thinking library(dplyr) thinking_residuals <- thinking |> mutate(residual = residuals(pay_model)) top_positive <- thinking_residuals |> arrange(desc(residual)) |> slice_head(n = 3) |> select(occupation, importance, annual_mean_wage, residual) top_negative <- thinking_residuals |> arrange(residual) |> slice_head(n = 3) |> select(occupation, importance, annual_mean_wage, residual) print("Largest Positive Residuals (Overpaid relative to skill model):") print(top_positive) print("Most Negative Residuals (Underpaid relative to skill model):") print(top_negative) # Find occupations above and below the model's predictions.`
2. `thinking model <- lm(pay ~ importance, data = jobs) jobs2 <- jobs |> mutate(residual = resid(model)) jobs2 |> select(occupation, importance, pay, residual) |> slice_max(residual, n = 3) jobs2 |> select(occupation, importance, pay, residual) |> slice_min(residual, n = 3) # Find occupations above and below the model's predictions.`

Student responses

1. `thinking pos_resid <- occupations |> mutate(residual = resid(model)) |> select(occupation, importance, pay, residual) |> slice_max(residual, n = 3) neg_resid <- occupations |> mutate(residual = resid(model)) |> select(occupation, importance, pay, residual) |> slice_min(residual, n = 3) pos_resid neg_resid # Find occupations above and below the model's predictions.`
2. `thinking thinking_residuals <- thinking |> mutate(residual = pay - predict(pay_model, newdata = thinking)) thinking_residuals |> slice_max(residual, n = 3, with_ties = FALSE) |> select(occupation, importance, pay, residual) thinking_residuals |> slice_min(residual, n = 3, with_ties = FALSE) |> select(occupation, importance, pay, residual) # Find occupations above and below the model's predictions.`

Student responses

1. `thinking_with_residuals <- thinking |> mutate(residual = residuals(pay_model)) # 3 largest positive residuals thinking_with_residuals |> arrange(desc(residual)) |> slice_head(n = 3) |> select(occupation, importance, pay, residual) # 3 most negative residuals thinking_with_residuals |> arrange(residual) |> slice_head(n = 3) |> select(occupation, importance, pay, residual) # Find occupations above and below the model's predictions.`
2. `thinking <- skills |> inner_join(wages, by = "occupation_code") |> filter(skill == "Critical Thinking") |> mutate(residual = pay - predict(pay_model)) |> top_n(3, residual) # Find occupations above and below the model's predictions.`
3. `thinking <- thinking |> mutate(residual = pay - predict(pay_model)) thinking |> arrange(desc(residual)) |> select(occupation, importance, pay, residual) |> slice_head(n = 3) thinking |> arrange(residual) |> select(occupation, importance, pay, residual) |> slice_head(n = 3)`

Student responses

1. `thinking <- thinking |> mutate(residual = pay - predict(pay_model)) # Find occupations above and below the model's predictions.`
2. `thinking <- thinking |> mutate(residual = pay - predict(pay_model)) |> arrange(desc(residual)) thinking # Find occupations above and below the model's predictions.`
3. `thinking <- thinking |> mutate(residual = pay - predict(pay_model)) |> arrange(desc(residual))|> slice_head(3) thinking`
4. `thinking <- thinking |> mutate(residual = pay - predict(pay_model)) |> top_n(3, residual) # Find occupations above and below the model's predictions.`
5. `thinking <- thinking |> mutate(residual = pay - predict(pay_model)) |> arrange(desc(residual)) |> group_by(skill) |> select(occupation, importance, pay, residual) |> top_n(3) # Find occupations above and below the model's predictions.`
6. `thinking <- thinking |> mutate(residual = pay - predict(pay_model)) thinking |> arrange(desc(residual)) |> select(occupation, importance, pay, residual) thinking |> arrange(residual) |> select(occupation, importance, pay, residual) # Find occupations above and below the model's predictions.`

Student responses

1. `thinking <- thinking |> mutate(residual = pay - predict(pay_model)) |> arrange(desc(residual)) |> slice_head(n = 3) # Find occupations above and below the model's predictions.`
2. `thinking <- thinking |> mutate(residual = pay - predict(pay_model)) |> top_n(3, residual) # Find occupations above and below the model's predictions.`
3. `thinking <- thinking |> mutate(residual = pay - predict(pay_model)) |> arrange(desc(residual)) |> slice_head(n = 3) thinking # Find occupations above and below the model's predictions.`
4. `thinking <- thinking |> mutate(residual = pay - predict(pay_model)) |> arrange(desc(residual)) |> slice_head(n=3) # Find occupations above and below the model's predictions.`
5. `thinking |> mutate(residual = pay- predict(pay_model)) |> top_n(3, residual) # Find occupations above and below the model's predictions.`
6. `thinking |> arrange(desc(residual)) |> head(3) |> select(occupation, importance, pay, residual) thinking |> arrange(residual) |> head(3) |> select(occupation, importance, pay, residual) # Find occupations above and below the model's predictions.`

Student responses

1. `thinking |> arrange(desc(residual)) |> select(occupation, importance, pay, residual) |> head(3)` thinking |> arrange(residual) |> select(occupation, importance, pay, residual) |> head(3) # Find occupations above and below the model's predictions.
2. `thinking |> arrange(desc(residual)) |> select(occupation, importance, pay, residual) |> slice_head(n = 3)` thinking |> arrange(residual) |> select(occupation, importance, pay, residual) |> slice_head(n = 3)
3. `thinking |> arrange(desc(residual)) |> slice_head(n = 3) |> select(occupation, importance, pay, residual)` thinking |> arrange(residual) |> slice_head(n = 3) |> select(occupation, importance, pay, residual) # Find occupations above and below the model's predictions.
4. `thinking |> slice_max(residual, n = 3) |> select(occupation, importance, pay, residual)` thinking |> slice_min(residual, n = 3) |> select(occupation, importance, pay, residual)

Student responses

1. `thinking |> mutate(residual = pay - predict(pay_model)) |> arrange(desc(residual)) |> slice_head(n = 3) # Find occupations above and below the model's predictions.`
2. `thinking |> arrange(desc(residual)) |> select(occupation, importance, pay, residual) |> head(3) thinking |> arrange((residual)) |> select(occupation, importance, pay, residual) |> head(3) # Find occupations above and below the model's predictions.`
3. `thinking |> arrange(desc(residual)) |> select(occupation, importance, pay, residual) |> slice_head(n = 3) thinking |> arrange(residual) |> select(occupation, importance, pay, residual) |> slice_head(n = 3) # Find occupations above and below the model's predictions.`
4. `thinking |> mutate(residual = pay - predict(pay_model)) |> arrange(desc(residual)) |> select(occupation, importance, pay, residual) |> slice_head(n=3) thinking |> arrange(residual) |> select(occupation, importance, pay, residual) |> slice_head(n=3) # Find occupations above and below the model's predictions.`
5. `thinking |> arrange(desc(pay)) |> head(3) # Find occupations above and below the model's predictions.`

Student responses

1. `thinking |> arrange(desc(residual)) |> head(3) |> select(occupation, importance, pay, residual) thinking |> arrange(residual) |> head(3) |> select(occupation, importance, pay, residual) # Find occupations above and below the model's predictions.`
2. `thinking |> arrange(desc(residual)) |> head(3) |> select(occupation, importance, pay, residual) thinking |> arrange(residual) |> head(3)|> select(occupation, importance, pay, residual) # Find occupations above and below the model's predictions.`
3. `thinking |> mutate(residual = pay - predict(pay_model)) |> arrange(desc(residual)) |> select(occupation, importance, pay, residual) |> slice_head(n = 3) thinking |> arrange(residual) |> select(occupation, importance, pay, residual) |> slice_head(n = 3) # Find occupations above and below the model's predictions.`
4. `thinking_new = thinking |> mutate(residual = pay - predict(pay_model)) |> select(occupation, importance, pay, residual) |> arrange(desc(residual)) |> slice_head(n = 3) |> print() # Find occupations above and below the model's predictions.`

Student responses

1. `thinking_residuals <- thinking |> mutate(residual = pay - predict(pay_model, newdata = thinking)) thinking_residuals |> arrange(desc(residual)) |> select(occupation, importance, pay, residual) |> head(3) thinking_residuals |> arrange(residual) |> select(occupation, importance, pay, residual) |> head(3) # Find occupations above and below the model's predictions.`
2. `thinking<- thinking|> mutate(residual = pay-predict(pay_model))|> arrange(desc(residual))|> top_n(3, residual) # Find occupations above and below the model's predictions.`
3. `thinking<- thinking|> mutate(residual = pay - predict(pay_model))|> top_n(3, residual) # Find occupations above and below the model's predictions.`
4. `thinking|>arrange(desc(residual))|> select(occupation,importance, pay, residual)|> slice_head(n = 3) thinking|>arrange((residual))|> select(occupation,importance, pay, residual)|> slice_head(n = 3) # Find occupations above and below the model's predictions.`

Student responses

1. `top_pos <- skills %>% arrange(desc(residual)) %>% select(occupation, importance, pay, residual) %>% slice_head(n = 3)`
`top_neg <- skills %>% arrange(residual) %>% select(occupation, importance, pay, residual) %>% slice_head(n = 3)`
`top_pos top_neg # Find occupations above and below the model's predictions.`
2. `top_positive <- occupations |> select(occupation, importance, pay, residual) |> slice_max(order_by = residual, n = 3)`
`top_negative <- occupations |> select(occupation, importance, pay, residual) |> slice_min(order_by = residual, n = 3)`
`top_positive top_negative # Find occupations above and below the model's predictions.`
3. `top_three <- thinking |> mutate(residual = pay - predict(pay_model)) |> arrange(desc(residual)) |> slice_head(n = 3)`
`|> select(occupation, importnace, pay, residual)`
`bottom_three <- thinking |> mutate(residual = pay - predict(pay_model)) |> arrange(residual) |> slice_head(n = 3)`
`|> select(occupation, importnace, pay, residual)`
`bottom_three top_three # Find occupations above and below the model's predictions.`