

Political outlook of young adults

Week 9, Lecture 15

<https://live.ds306.org>

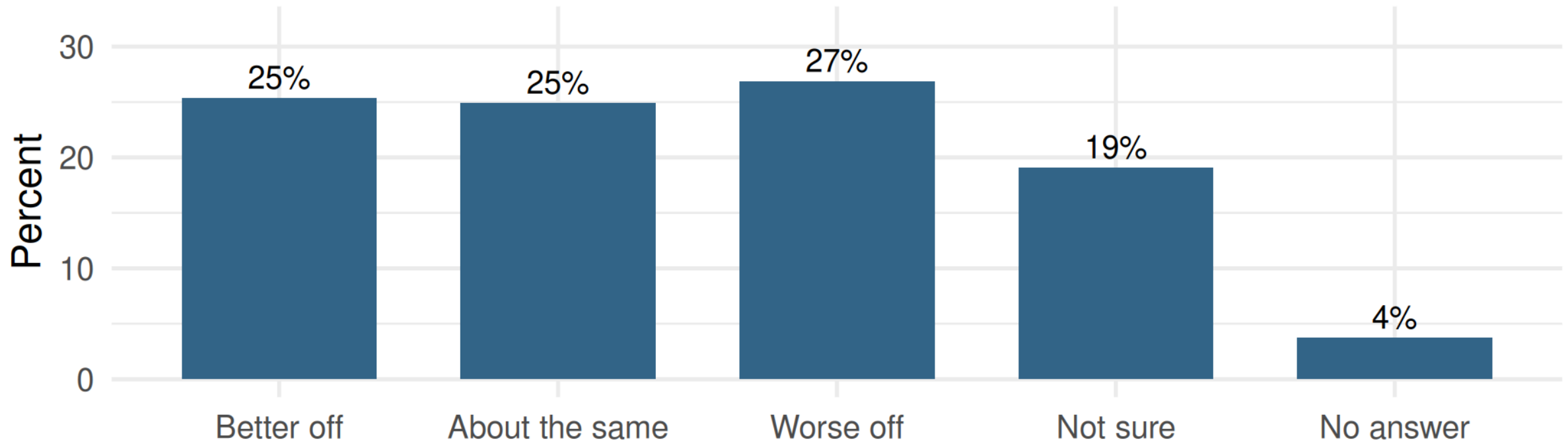
October 27, 2026

Will you be better off than your parents?

When you reach your parents' age, do you expect to be financially better off, about the same, or worse off than they are now?

Will we be better off than our parents?

Harvard asked young adults how they expect to do financially at their parents' age.



Only about one in four expect to be better off.

What about AI and jobs?

What percentage of 18–29-year-olds call AI replacing jobs a serious threat to their long-term financial security? (Enter a number from 0 to 100.)

The Harvard Youth Poll

```
list.files("data/harvard")
```

- 2,024 young adults, ages 18–29, surveyed September 30–October 7, 2026.
- Questions about financial security, politics, and public policy.
- Each CSV file contains responses from one question.

[Harvard Youth Poll, Fall 2026](#)

What questions did they ask?

`items.csv` lists the questions, their type (`kind`), and the file containing each one's responses.

```
items <- read_csv("data/harvard/items.csv")

items |>
  select(item, kind, file) |>
  print(n = Inf)
```

Which questions would you most like to investigate?

Two questions for today

1. How concerned are young adults about AI replacing jobs?
2. Do these concerns cut across party lines?

Skills for today

- Combine data from many files into one table.
- Use **iteration** with a for loop and `map( )`.
- Join question types and summarize responses across issues.
- **Debug** code by checking what it produces.

Q1: How concerned are young adults about AI?

Load the AI survey

```
ai <- read_csv("data/harvard/threat_ai_jobs.csv")  
glimpse(ai)
```

- `item` identifies the survey question.
- `group` identifies all respondents or a party group.
- `response` gives the answer category.
- `percent` gives the weighted percentage choosing that answer within the group.
- `base_n` gives the number of respondents in the group.

How many call AI a serious threat?

What percentage of all respondents consider AI a “Very serious” or “Somewhat serious” threat?

▼ [Inspect the AI response categories]

```
ai |> filter(group == "All") |> select(response, per
```

```
# Calculate the overall serious-threat percentage.
```

- Calculate the percentage calling AI a serious threat among all respondents, combining Very serious and Somewhat serious.

Visualize the overall responses

▼ [Responses among all respondents]

```
overall <- ai |> filter(group == "All")
```

```
overall |>  
  ggplot(aes(response)) +  
  geom_bar()
```

Why are all the bars equal? Fix the plot to show the response percentages and submit it.

Do opinions about AI differ by political affiliation?

Investigate the response distributions for Democrats, Republicans, and independents.

▼ [Keep the three party groups]

```
by_party <- ai |> filter(group != "All")
```

```
# Investigate how the responses differ across parties.
```

- Make a table or plot showing the response percentages for the three party groups.
- Calculate the percentage calling AI a serious threat within each party group.

Is concern about AI independent of party?

We don't have the raw respondent-level data, only the summarized percentages, so estimate counts as $\text{round}(\text{base_n} * \text{percent} / 100)$. Omit Not sure and Refused.

▼ [Estimate counts for Democrats and Republicans]

```
party_counts <- ai |>
  filter(group %in% c("Democrat", "Republican"),
         !response %in% c("Not sure", "Refused")) |>
  mutate(estimated_n = round(base_n * percent / 100))
```

```
# Create a serious/not-serious column.
# Sum the counts into a table with:
# xtabs(estimated_n ~ serious + group, data = ...)
```

- Build a 2x2 table of estimated counts: serious/not serious by Democrat/Republican. Combine Very serious and Somewhat serious.
- Use `chisq.test()` to test independence at the 5% significance level.

Q2: Do these concerns cut across party lines?

AI was one of seven financial-threat questions. The survey also asked about nine policies.

Are young Democrats, Republicans, and independents worried about the same things? Do they support the same policies?

We need to bring the question files together to investigate.

Get all the data into one table

We have sixteen questions, each in its own file. We want to analyze them together. Here are a few rows from the combined table.

```
# A tibble: 4 × 5
  item                group response    percent base_n
  <chr>              <chr> <chr>      <dbl>   <dbl>
1 Large corporations All    Very serious    40.1    2024
2 Large corporations All    Somewhat serious 28.1    2024
3 AI replacing jobs  All    Very serious    35.7    2024
4 AI replacing jobs  All    Somewhat serious 28.7    2024
```

The columns stay the same. We stack the rows from each file.

Strategy 1: type out each filename

`bind_rows()` stacks tables, matching their columns by name.

```
survey_manual <- bind_rows(  
  read_csv("data/harvard/threat_billionaires.csv", show_col_types = FALSE),  
  read_csv("data/harvard/threat_corporations.csv", show_col_types = FALSE),
```

▼ [\[Read the other fourteen question files\]](#)

```
  read_csv("data/harvard/threat_immigration_jobs.csv", show_col_types = FALSE),  
  read_csv("data/harvard/threat_ai_jobs.csv", show_col_types = FALSE),  
  read_csv("data/harvard/threat_cost_of_living.csv", show_col_types = FALSE),  
  read_csv("data/harvard/threat_national_debt.csv", show_col_types = FALSE),  
  read_csv("data/harvard/threat_trade_wars.csv", show_col_types = FALSE),  
  read_csv("data/harvard/policy_medicare_for_all.csv", show_col_types = FALSE),  
  read_csv("data/harvard/policy_wealth_tax.csv", show_col_types = FALSE),  
  read_csv("data/harvard/policy_deportations.csv", show_col_types = FALSE),  
  read_csv("data/harvard/policy_israel_aid.csv", show_col_types = FALSE),  
  read_csv("data/harvard/policy_transgender_sports.csv", show_col_types = FALSE),  
  read_csv("data/harvard/policy_stock_trading.csv", show_col_types = FALSE),  
  read_csv("data/harvard/policy_voter_id.csv", show_col_types = FALSE),  
  read_csv("data/harvard/policy_abortion.csv", show_col_types = FALSE),  
  read_csv("data/harvard/policy_tariffs.csv", show_col_types = FALSE)
```

```
)
```

Use iteration

The filenames are already listed in `items.csv`. We will study ways to read them all efficiently.

▼ [Add the folder path to each filename in `items`]

```
survey_files <- paste0("data/harvard/", items$file)
```

```
survey_files
```

Strategy 2: a for loop

A for loop runs the code inside its braces once for each file. `file` takes the next filename on each pass.

```
for (file in survey_files) {  
  survey <- read_csv(  
    file, show_col_types = FALSE)  
}  
survey |> count(item)  
  
# Repair the loop to keep every question.
```

- Explain in a comment why this loop leaves only one question in survey.
- Repair the loop so it keeps every table, then combine them into survey.
- Check that survey contains sixteen

Keep each table the loop reads

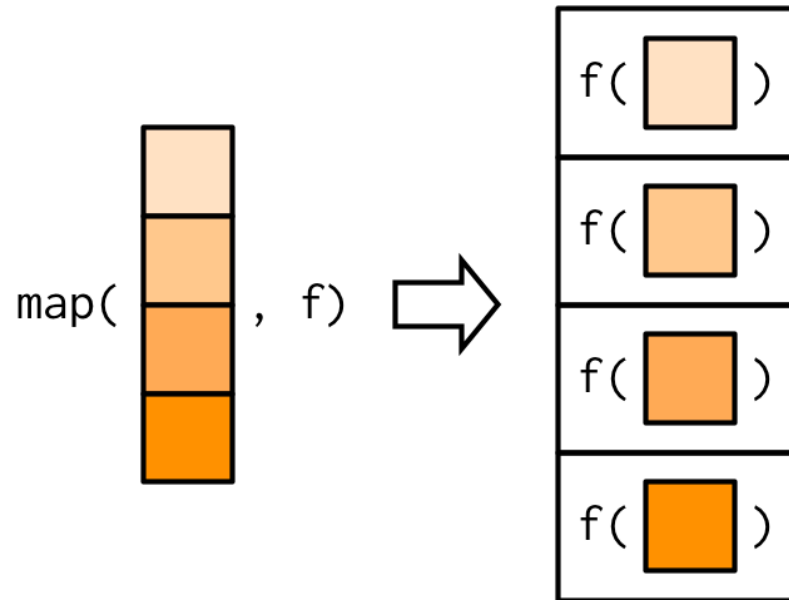
A **list** can hold all sixteen tables. `survey_tables[[file]]` stores a table under its filename.

```
survey_tables <- list()
for (file in survey_files) {
  survey_tables[[file]] <- read_csv(file, show_col_types = FALSE)
}
survey_loop <- list_rbind(survey_tables)
survey_loop |> count(item)
```

`list_rbind()` stacks the tables in the list into one data frame.

Apply a function to each element

Each colored box is an input. The same function f is applied to each one.



Our inputs are filenames, f is `read_csv( )`, and each output is a table.

Hadley Wickham, Advanced R

Strategy 3: map

`map()` calls a function on each input and collects its outputs in a list. Here the inputs are filenames and the function is `read_csv()`.

```
survey_tables <- map(survey_files, read_csv, show_col_types = FALSE)
survey_tables[[1]]
```

`[[1]]` displays the first table in the list. Which part of the loop is `map()` handling for us?

Collect all sixteen questions

Use `map()` and `list_rbind()` to create the combined table.

▼ [The sixteen question filenames]

```
survey_files
```

```
# Read every file and combine its rows into survey.
```

- Use `map` with `read_csv`, then `list_rbind`, to collect all sixteen files in `survey`.
- Check that each question has 24 rows and that there are 384 rows in total.
- Check that the AI rows match the ai

Check the combined table

```
survey <- survey_files |> map(read_csv, show_col_types = FALSE) |>  
  list_rbind()  
survey |> count(item)
```

Six response categories for four groups: 24 rows per question.

Separate threats and policies

items identifies seven financial-security questions and nine policy questions.

▼ [Attach the question type]

```
survey_with_types <- survey |>  
  left_join(items |> select(item, kind), by = "item")
```

▼ [Create a table for each question type]

```
threats <- survey_with_types |> filter(kind == "threat")  
policies <- survey_with_types |> filter(kind == "policy")
```

```
survey_with_types |> count(kind)
```

Use `threats` for questions about financial security and `policies` for questions about policy support.

What is your biggest financial worry?

Which of these do you feel is the biggest threat to your long-term financial security?

What is each party's biggest financial worry?

Among the seven questions in `threats`, which financial-security issue concerns each party most? Find which financial-security issue has the highest combined percentage of Very serious and Somewhat serious responses for each party.

▼ [Inspect the response categories]

```
threats |> count(response)
```

```
# Find the biggest concern within each party group.
```

What is the top financial concern among Democrats, Republicans, and independents? Report the concern and percentage for each group.

Find the top concern in each party

▼ [Keep the two serious-threat response categories]

```
threat_results <- threats |>
  filter(response %in% c("Very serious", "Somewhat serious")) |>
```

▼ [Add their percentages for each question and group]

```
group_by(item, group) |>
  summarize(percent = sum(percent), .groups = "drop")
```

```
threat_results |>
  filter(group != "All") |>
  group_by(group) |>
  top_n(1, percent)
```

Cost of living tops the list for all three groups.

Which concerns divide Democrats and Republicans?

Define **net serious** as the percentage saying Very serious or Somewhat serious minus the percentage saying Not very serious or Not a threat.

How do party groups differ on this score?

▼ [The four threat-response categories]

```
threats |>
  filter(response %in% c("Very serious", "Somewhat s
                        "Not very serious", "Not a
```

```
# Calculate net-serious scores and find the largest par
```

- Calculate net_serious for each concern and group in threat_net; Not sure and Refused contribute zero.
- Put the party scores side by side

Ranking the party differences

▼ [Add serious responses and subtract not-serious responses]

```
threat_net <- threats |>
  filter(response %in% c("Very serious", "Somewhat serious",
                        "Not very serious", "Not a threat")) |>
  mutate(contribution = ifelse(
    response %in% c("Very serious", "Somewhat serious"), percent, -percent)) |>
  group_by(item, group) |>
  summarize(net_serious = sum(contribution), .groups = "drop")
```

▼ [Put the party scores side by side]

```
threat_comparison <- threat_net |>
  pivot_wider(names_from = group, values_from = net_serious) |>
```

▼ [Calculate the Democratic-minus-Republican gap]

```
mutate(party_gap = Democrat - Republican, abs_gap = abs(party_gap))
```

```
threat_comparison |>
  arrange(desc(party_gap)) |>
  select(item, Democrat, Republican, Independent, party_gap)
```

Check the ranking

```
threat_comparison |>  
  arrange(desc(abs_gap)) |>  
  select(item, Democrat, Republican, Independent, abs_gap)
```

Immigration is near the top because the Republican score is much higher.

What concerns are shared? Which are divisive?

Make a plot of the net-serious scores for each issue across dems, republicans, and independents.

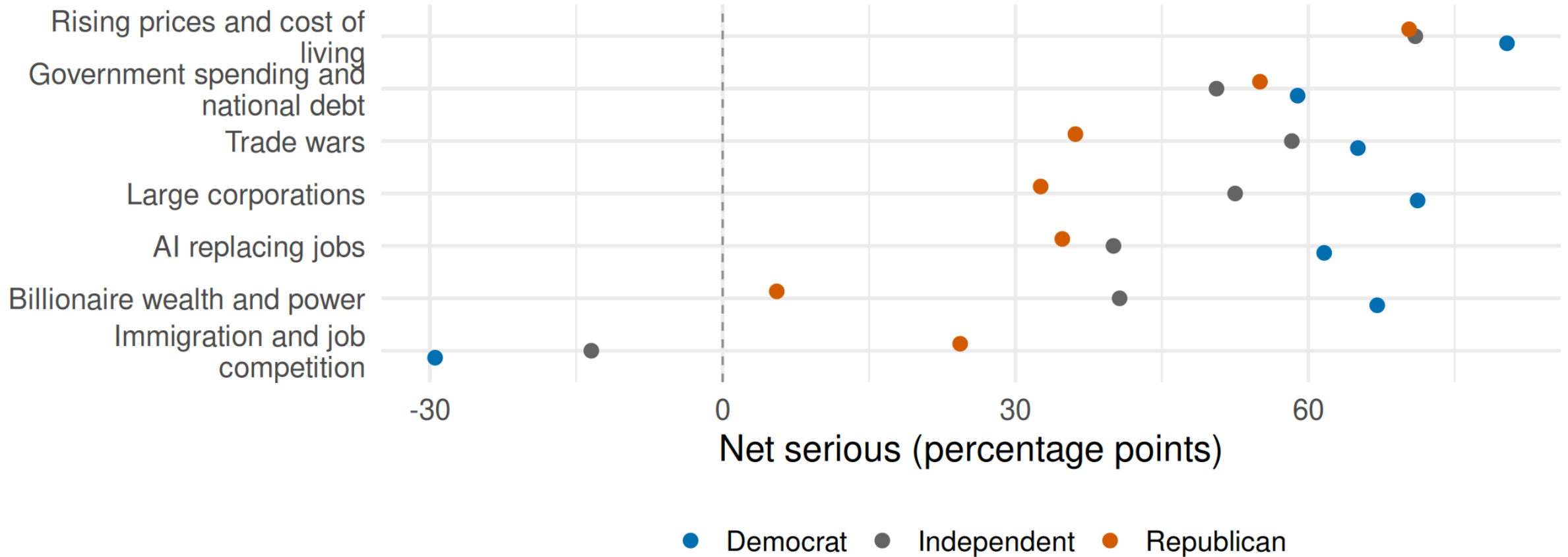
▼ [Net-serious scores for all concerns]

```
threat_net
```

```
# Plot the scores for each party group.
```

Plot net-serious scores for all seven concerns by party. Identify one threat that is widely shared, and one that is divisive.

What does the full set show?



Which policies win a majority in every party?

- What proposals are supported by more than 50% of Democrats, Republicans, and independents?
- The questions include healthcare, taxes, stock trading, voting rules, immigration, abortion, school sports, military aid, and tariffs.

Which policy do you predict will have majority support among Democrats, Republicans, and independents? What makes you think so?

Find the policies with support across parties

Calculate support for all nine policies, then find those that pass the 50% threshold in all three party groups.

▼ [Inspect the policy response categories]

```
policies |> count(response)
```

```
# Calculate policy_results and find majority support ac
```

- Combine Strongly support and Somewhat support for each policy and group; save the percentages as policy_results.
- Find policies with more than 50% support in each of the three party

Can support be greater than 100%?

```
policy_attempt <- policies |>
  filter(response %in% c("Strongly support", "Somewhat support")) |>
  group_by(group) |>
  summarize(percent = sum(percent), .groups = "drop")
policy_attempt
```

Why are these support percentages greater than 100? Explain what went wrong and how to repair the calculation.

Calculate support within each group

▼ [Keep the two support categories]

```
policy_results <- policies |>  
  filter(response %in% c("Strongly support", "Somewhat support")) |>
```

▼ [Add their percentages for each policy and group]

```
  group_by(item, group) |>  
  summarize(percent = sum(percent), .groups = "drop")
```

```
policy_results |> count(item)
```

Check which policies passed

▼ [Put the group support percentages side by side]

```
policy_comparison <- policy_results |>
  pivot_wider(names_from = group, values_from = percent) |>
```

▼ [Calculate the party gap]

```
  mutate(party_gap = Democrat - Republican, abs_gap = abs(party_gap))
```

```
policy_comparison |>
  filter(Democrat > 50, Republican > 50, Independent > 50) |>
  select(item, Democrat, Republican, Independent)
```

Did either result surprise you?

Which policies have the biggest and smallest party gaps?

▼ [The support percentages and party gaps]

```
policy_comparison
```

```
# Find the largest and smallest absolute party gaps.
```

Which policies have the biggest and smallest Democrat-Republican gaps in support?

Congressional stock trading

The proposal would prevent members of Congress and their spouses from trading individual stocks while in office.

```
# A tibble: 1 × 5
  item                All Democrat Republican Independent
  <chr>                <dbl>      <dbl>         <dbl>         <dbl>
1 Congressional stock-trading ban 63.3      70.7          62.4          62.4
```

A majority supports it in each party group. It also has the smallest Democratic–Republican gap among these nine policies.

Where do the parties disagree?

Name a policy on which Democrats and Republicans strongly disagree.

Popular but controversial policies?

Which policies have majority support overall but fall below 50% in at least one party group?

▼ [Overall and party support for every policy]

```
policy_comparison
```

```
# Find policies that meet the condition above.
```

- Find every policy with more than 50% support overall and less than 50% support in at least one party group.
- Show the overall and party percentages, and explain one

Which policies did you find?

```
policy_comparison |>
  filter(All > 50,
         Democrat < 50 | Republican < 50 | Independent < 50) |>
  select(item, All, Democrat, Republican, Independent)
```

Do independents track Democrats or Republicans?

We have each group's support for nine policies.

Across those policies, does the pattern among independents resemble Democrats' or Republicans' more closely?

Use scatterplots and correlation to investigate.

Investigate independent opinion

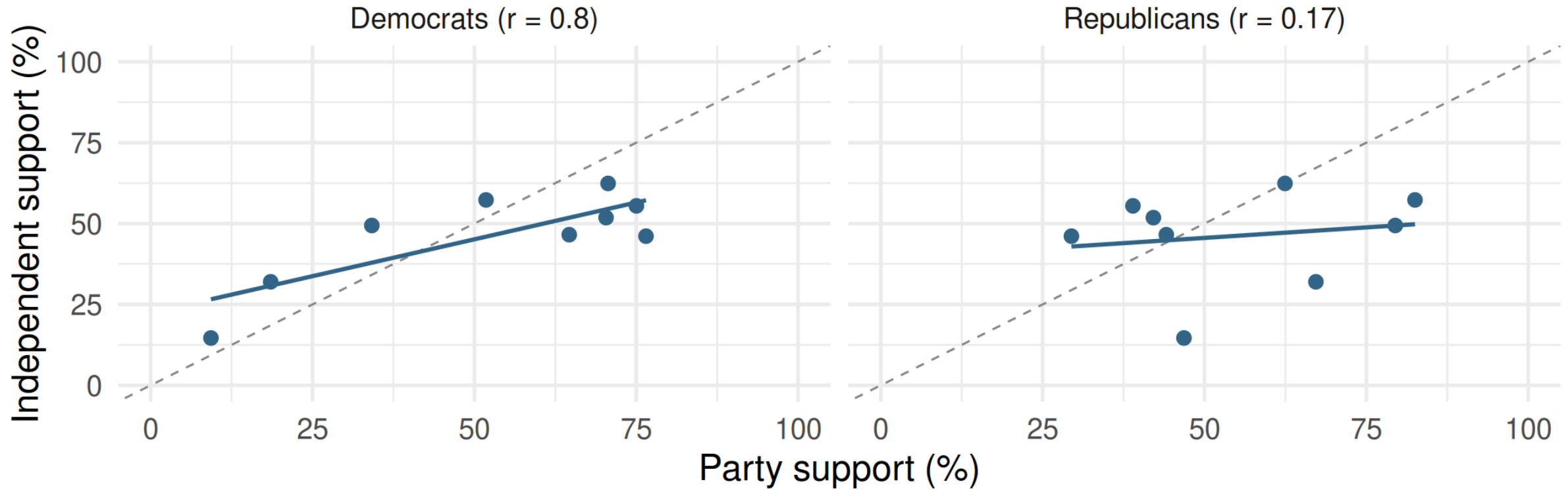
▼ [Prepare two plotting panels]

```
independent_plot_data <- policy_comparison |>
  select(item, Independent, Democrat, Republican) |>
  pivot_longer(c(Democrat, Republican),
               names_to = "party", values_to = "supp
```

```
# Compare each party with independents.
```

- Plot Independent against each party's support, with one point per policy.
- Use `policy_comparison` to calculate both correlations with `cor()`.
- Explain which party's pattern tracks independents more

Which pattern is more similar?



Does higher correlation mean closer agreement?

▼ [Inspect support for each policy]

```
policy_comparison |>
  select(item, Democrat, Independent, Republican)
```

```
# Find a counterexample.
```

Name a policy where independents are closer to Republicans. Give the three parties' support percentages.

What we practiced

- Read a collection of files with a loop or `map()`.
- Combine the returned tables with `list_rbind()`.
- Join question types from `items` and separate threats from policies.
- Use scatterplots and correlation to study party agreement.
- Communicate a finding with a table, plot, and explanation.