Major Disaster Declarations Have Turned Partisan: How Do We Know, and What's the Fix?

George Doehne

I. Introduction

In the aftermath of overwhelming natural disasters, states may request a federal major disaster declaration to get federal relief, but their issuance is subject to presidential discretion. This brief examines whether partisan bias affects declaration issuance. The data show there have been major partisan differences in major disaster declaration request outcomes during the current administration. The nature of these changes helps inform policy approaches for reforming the major disaster declaration process.

II. Background

The modern major disaster declaration process dates back to the passage of the Robert T. Stafford Disaster Relief and Emergency Assistance Act, or Stafford Act, by Congress in 1988. Title IV of the Act codified a specific procedure for major disaster declarations: when states[1] suffer natural disasters that outmatch their own resources for recovery, their governor may formally request assistance from the President of the United States via the Federal Emergency Management Agency, or FEMA. Within each request, states provide evidence for the overwhelming nature of the disaster, specify which counties they are seeking assistance for, and ask for specific FEMA programs to be activated. These consist of Individual Assistance programs (which include counseling, legal services, and various forms of direct financial support to individuals and households) and Public Assistance programs (which provide aid to state and local governments, including the repair or replacement of public facilities and services, e.g., debris removal or road repairs).[2]

Upon receiving a state's request, FEMA considers many factors in deciding whether to recommend to the president that it be granted or denied. These include the per capita cost of assistance, the amount of insurance coverage in affected areas, the recent disaster history, and the state's own fiscal capacity. Whatever FEMA's recommendation, the power to issue a declaration lies solely with the president, who may choose to reject a declaration request for any reason. A rejected request may be appealed only once.[3]

III. Major Disaster Declarations In Context

Between January 20, 1989—the beginning of the first full post-Stafford Act presidency—and May 1, 2026, there have been 2,357 major disaster declaration requests, per FEMA's National Emergency Management Information System. Of these, 1,940 (or about 82 percent) were granted and 417 (18 percent) were denied. The average wait time for a presidential decision

[1] Federally recognized Native tribes, U.S. territories, and Freely Associated States may also request Major Disaster Declarations, but are excluded from analyses and figures in this brief.
[2] "How a Disaster Gets Declared," FEMA.Gov, July 29, 2025, https://www.fema.gov/disaster/how-declared.
[3] FEMA.Gov, "How a Disaster Gets Declared."

was 15.2 days and the median wait time was 10 days. Figure 1 below shows that the number of requests per year has not shown a noticeable trend upward or downward since 1989, and neither has the number of denied requests as a subset of that group. Nor is there any visible pattern to the number of requests during the four Republican presidential administrations versus the three Democratic ones.

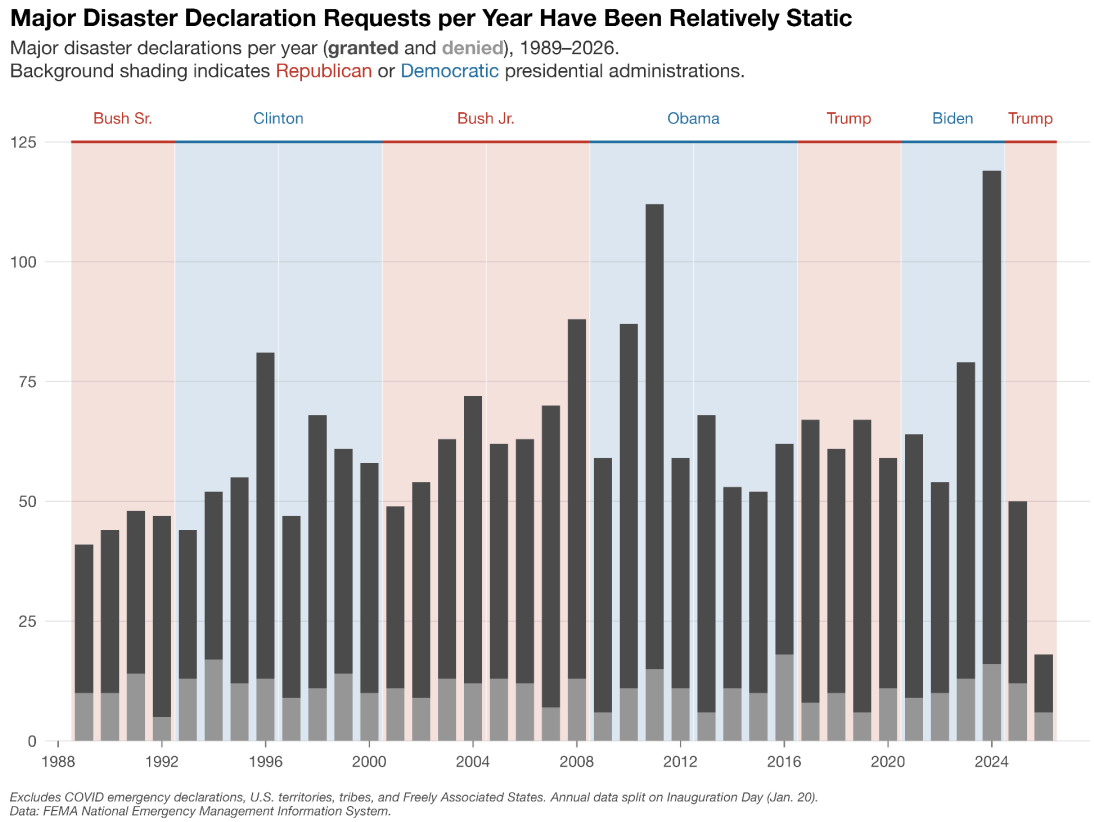


Figure 1

Figure 2 below shows the geographic distribution of disaster declarations by category during that same 1989–2026 period. The distribution varies much more by the type of disaster than by quantity. The Mountain West states have by far the fewest declarations, followed by the smaller northeastern states. After aggregating FEMA data's disaster types into a smaller and less redundant set of categories, it becomes clear that severe storms (which include tornadoes, hail, straight-line winds, etc.) dominate disaster declarations in most states. It is not as if California only requests assistance for earthquakes and Florida only for hurricanes; states are generally more alike than different.

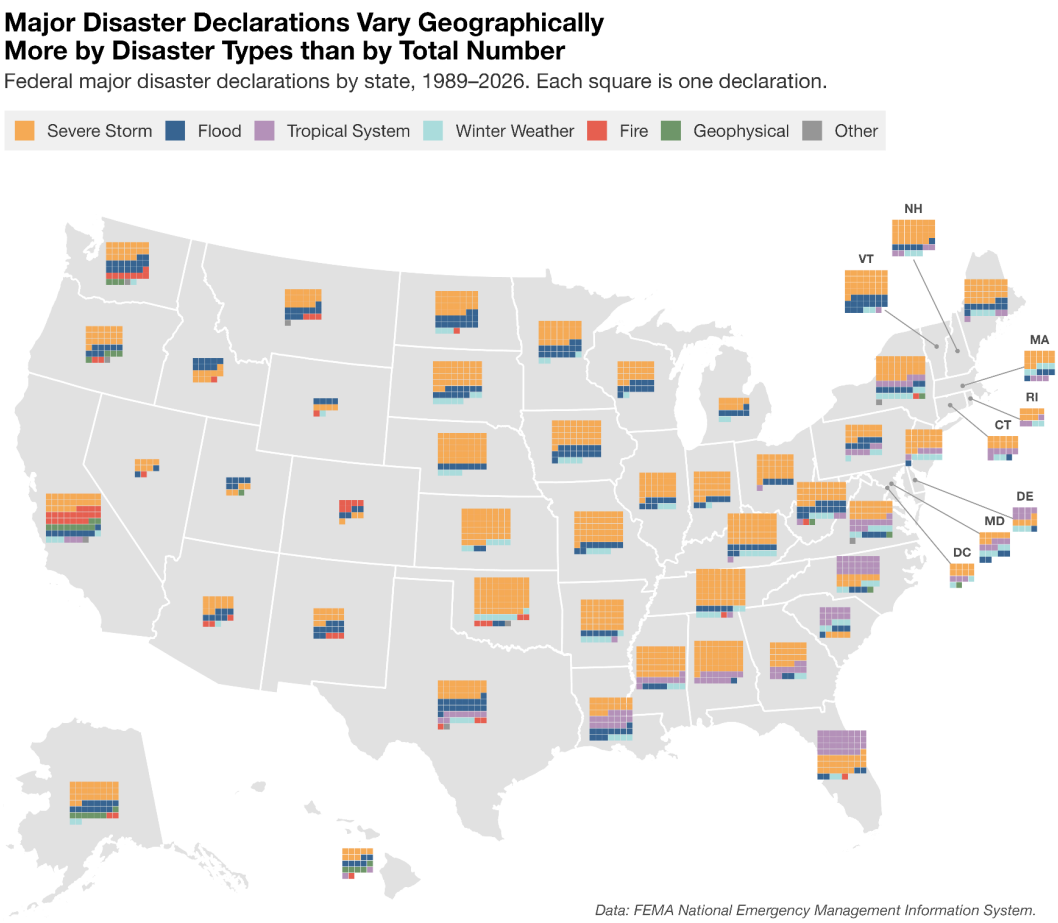
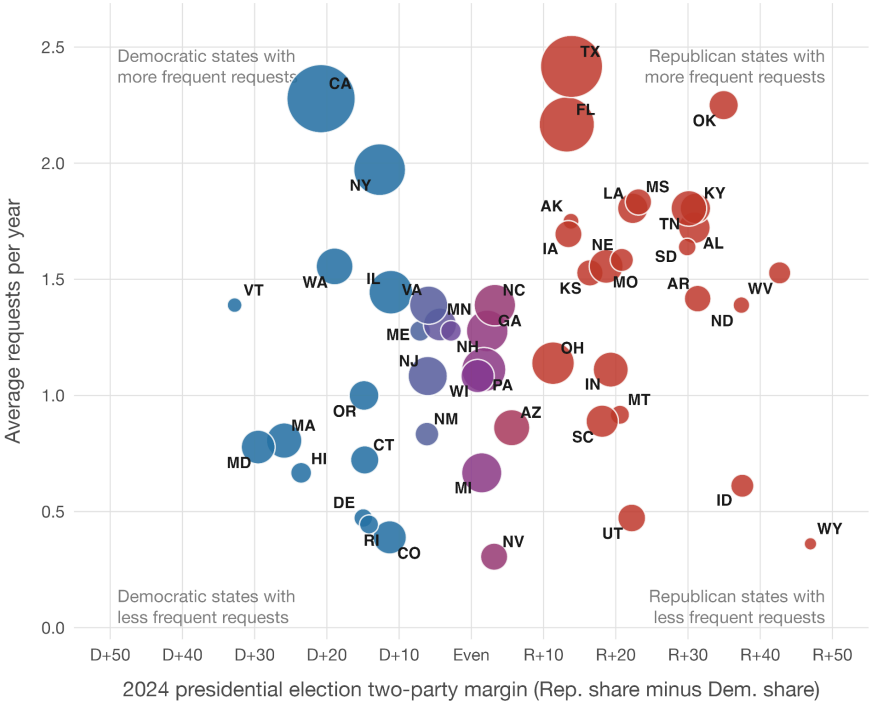


Figure 2

Figure 3 below directly plots the relationship between modern state partisanship (as defined by 2024 presidential election vote margin) and historic disaster declaration frequency. More Republican states frequently request major disaster declarations than Democratic states: 15 Republican states request a declaration 1.5 times per year or more on average, but only three Democratic states do. If it were the opposite, and Democratic states were more frequent requesters, we might not be surprised to see more rejections, or a lower percentage of granted requests, for Democratic states. But this is not the case, and is worth bearing in mind for the next section.

States' Partisan Leans Are Somewhat Predictive of Past Major Disaster Declaration Request Frequency

Average number of major disaster declaration requests per year, 1989–2024. Each dot is a state, sized by 2024 population and colored by 2024 two-party presidential election margin.



Excludes COVID emergency declarations, U.S. territories, tribes, and Freely Associated States. Annual data split on Inauguration Day (Jan. 20).
Data: FEMA National Emergency Management Information System, U.S. Census Bureau, Fox News Election Results.

Figure 3

IV. Major Disaster Declarations by Partisanship

Knowing the temporal and geographic distribution of past disaster declaration requests and decisions is not strongly correlated with state partisanship makes identifying more recent partisan patterns easier. In an effort to avoid using a metric for partisanship that biases the outcome, I employ three different criteria for Democratic versus Republican states. The first is presidential vote outcome: during any presidential term, a state that was won by the Democratic candidate in the corresponding presidential election is blue, and for the Republican candidate, red. The second definition is by governor's party: when the state has a Democratic governor it is blue, and when it has a Republican governor, red. The third is by the party of the majority of three officeholders, the governor and both senators: when two or more of those offices are held by a Democrat the state is blue, and when held by a Republican, red.[4]

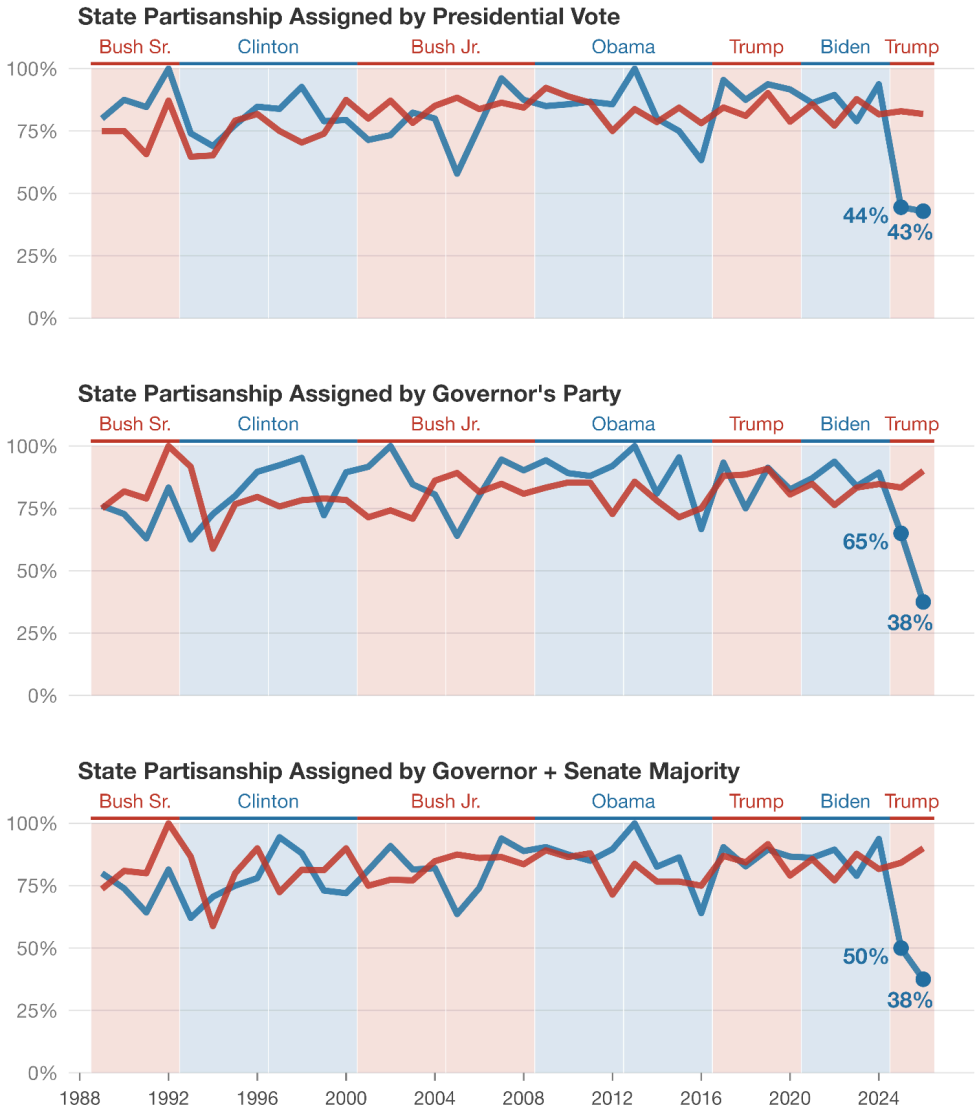
Applying these criteria to FEMA data on granted and denied declarations, Figure 4 below shows the average time span in days between a gubernatorial request for a presidential disaster declaration and the president's decision, calculated separately for Democratic vs. Republican states. It shows that grant rates for states' disaster declaration requests held

[4] Independent or third-party office-holders are assigned to the party they caucus with, and are otherwise excluded.

steady at or above 75 percent from 1989 to 2024. Never, for states belonging to either party, did the grant rate drop below 50 percent. However, in the first two years of the second Trump administration, the Democratic state grant rate has plummeted while the Republican state grant rate has remained above 80 percent. This change is evident no matter the methodology. Under all three definitions, there is a long period of consistent and stable grant rates followed by an abrupt decline for Democratic states in the last two years.

By Any Metric, Major Disaster Declaration Grant Rates Are Diverging by Party

Overall major disaster declaration grant rate, by year, for **Republican states** vs. **Democratic states**, 1989–2026. Background shading indicates Republican or Democratic presidential administrations.



Excludes COVID emergency declarations, U.S. territories, tribes, and Freely Associated States. Annual data split on Inauguration Day (Jan. 20).
Data: FEMA National Emergency Management Information System, MIT Election Data, Ballotpedia, @unitedstates/congress-legislators GitHub.

Figure 4

Using presidential election votes as our litmus test for partisanship, Figure 5 below shows the grant rate for requests by all Republican states under Democratic presidents, and for all Democratic states under Republican presidents from 1989 until Trump's second term. Next to those two columns are the grant rates for Republican and Democratic states under Trump. Again, in 2025–2026 Republican states' grant rates are unremarkable but Democratic states' have dropped sharply.

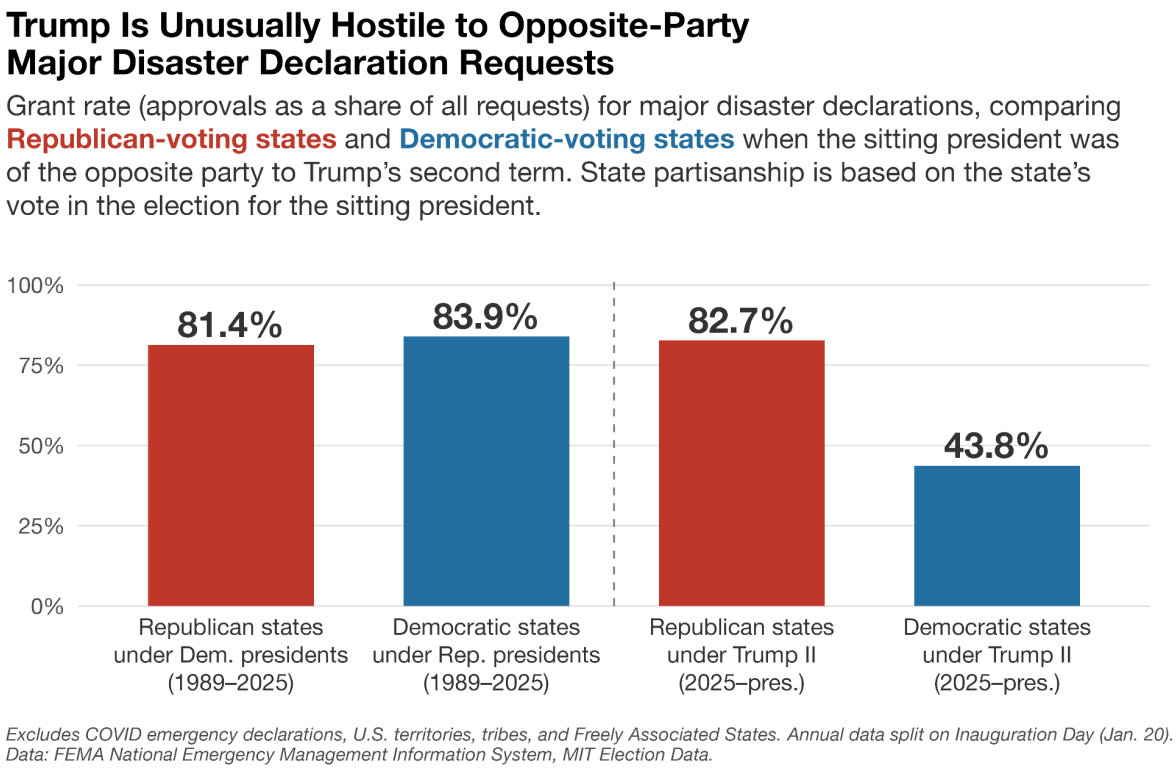


Figure 5

A. Wait Times

Another outcome we can use to check for biases is time between a state's request and the president's decision thereupon. There are effectively no hard-stop time restrictions within the major disaster declaration request process. Only two time limits apply: governors have 30 days from the disaster to submit their request for a declaration, and 30 days from the denial of their request to appeal. Both of these time limits may be extended by submitting a written request with justification to FEMA. Neither of these restrictions are statutory; they originate not from the Stafford Act but from FEMA implementing regulations.[5] There are no time restrictions whatsoever on FEMA or the president's deliberation or responses to a request or appeal.

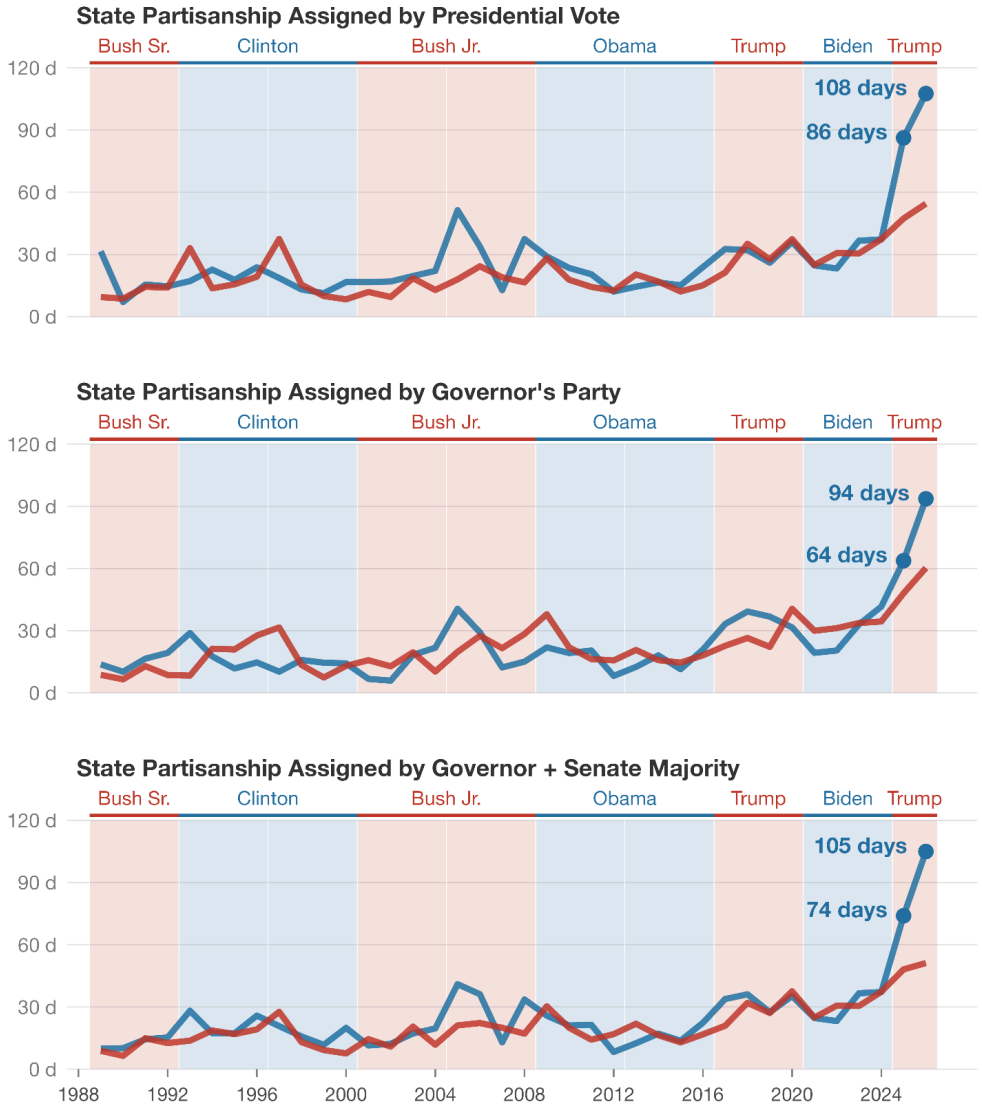
Since the time span between a governor's request and the president's answer is totally subject to the president's discretion, it is one clear way to see whether that discretion is being

[5] "44 CFR 206.36 -- Requests for Major Disaster Declarations.," FEMA.Gov, https://www.ecfr.gov/current/title-44/part-206/section-206.36.

exercised uniformly. Figure 6 below shows, for each year between 1989–2026, the average number of days between a governor's declaration request and the president's decision. Again, no matter the method used to assign Democratic versus Republican states, wait times for states of either party were consistent from 1989 until 2024. Then, in 2025, wait times for all states increased; they have not quite doubled for Republican states but more than tripled for Democratic states.

By Any Metric, Major Disaster Declaration Decision Wait Times Are Diverging by Party

Average wait, by year, for major disaster declaration decisions for **Republican states** vs. **Democratic states**, 1989–2026. Background shading indicates Republican or Democratic presidential administrations.



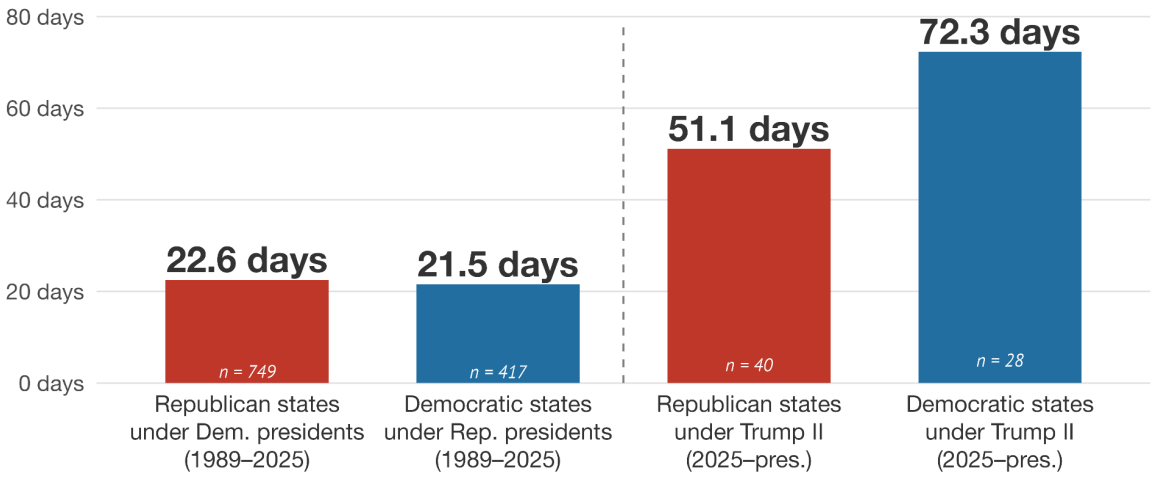
Excludes COVID emergency declarations, U.S. territories, tribes, and Freely Associated States. Annual data split on Inauguration Day (Jan. 20).
Data: FEMA National Emergency Management Information System, MIT Election Data, Ballotpedia, @unitedstates/congress-legislators GitHub.

Figure 6

From 1989 to 2025, e.g., until the second Trump administration, the average length of time between a governor's request and the president's decision was ___ days. For Republican states' requests under Democratic presidents, and for Democratic states' requests under Republican presidents, the mean time to respond was just over three weeks. During the second Trump administration, the average time until a decision is made for a Republican state's request has more than doubled to over seven weeks, and for a Democratic state's request the average has more than tripled to over ten weeks.

Trump Is Unusually Slow to Answer Major Disaster Declaration Requests

Average days from request to decision for major disaster declarations, comparing **Republican-governed states** and **Democratic-governed states** when the sitting president was of the opposite party. State partisanship is based on the party of the sitting governor on the decision date.



Excludes COVID emergency declarations, U.S. territories, tribes, and Freely Associated States. Annual data split on Inauguration Day (Jan. 20). Data: FEMA National Emergency Management Information System, MIT Election Data.

Figure 7

V. Summary

This data is consistent with reporting that has connected specific major disaster declaration grants and denials under the second Trump administration to specific political alignments. It is worth highlighting that, despite additional reporting that President Trump evinced a desire to wield the major disaster declaration power to reward allies and punish rivals during his first term,[67] the data does not show that he followed through with it until he regained office in 2025.

The administration's explanation for these changes are that they are part of a an effort to have

[6] Thomas Frank and Scott Waldman, "Trump Ignored Disaster Aid Request from Political Rival," *E&E News*, October 24, 2024, https://www.eenews.net/articles/trump-ignored-disaster-aid-request-from-political-rival/.
[7] Scott Waldman and Thomas Frank, "Trump Initially Refused to Give California Wildfire Aid Because It's a Blue State, Ex-Aide Says," *E&E News*, October 3, 2024, https://www.eenews.net/articles/helene-isnt-the-first-time-trump-inserted-politics-into-a-natural-disaster/.

"The Federal Government [focus] its support on truly catastrophic disasters—massive hurricanes, devastating earthquakes, or wide-scale attacks on the homeland",[8] per a National Security Council spokesperson. This rationale is not borne out by the data either.

Figure 8 below shows two metrics: the top plot shows the median number of counties per disaster declaration, and the bottom the mean number of FEMA programs approved per declaration annually. If the federal government had raised the bar to only grant declarations for disasters of unusual scope and severity, we would expect the median number of counties per declaration to be higher than before and the mean number of programs to also have risen. Instead, we see the opposite: there is no detectable rise in the number of counties per declaration, and the number of FEMA programs activated per declaration has instead dropped.

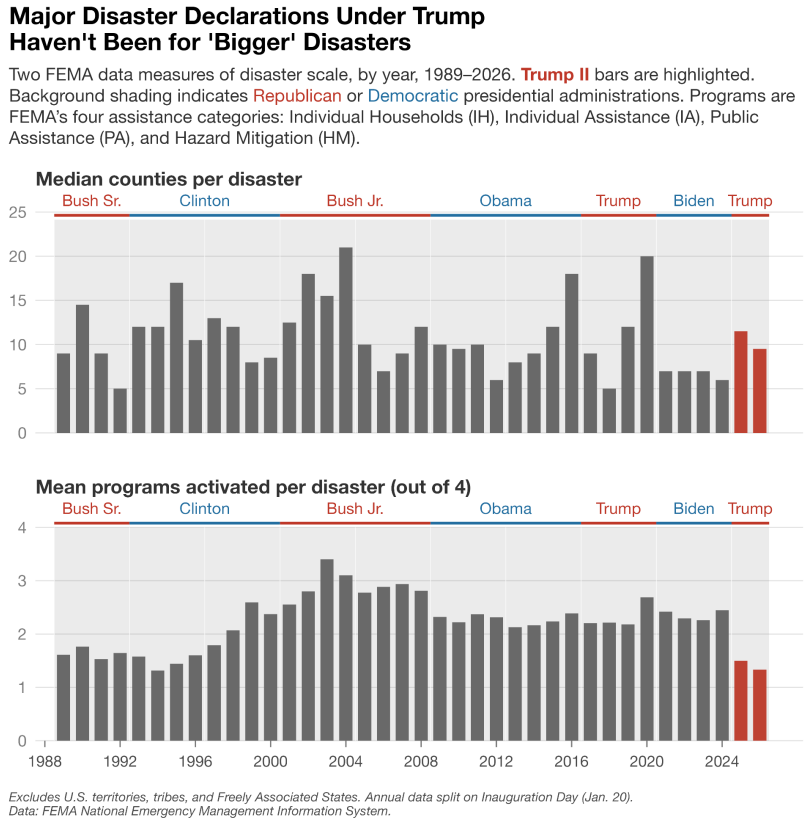


Figure 8

The data make it clear that the second Trump administration has overseen drastic changes in major disaster declaration request outcomes that are especially divergent by state partisanship. Republican states hear back from the federal government about their major disaster declaration requests in twice the time they used to, but it is still, four times out of five, a grant of their request. Democratic states hear back from the government after more than three times the timespan they used to and more often than not it is a rejection. It is difficult to escape the conclusion that the current administration is putting a thumb on the scale.

[8] Alex Brown, "Trump Denies Disaster Aid, Tells States to Do More," Stateline, April 25, 2025, https://stateline.org/2025/04/25/trump-denies-disaster-aid-tells-states-to-do-more/.

VI. Recommendations

These changes take place against a backdrop of potential executive reform to FEMA's role in federal disaster relief. In early 2025 the Trump administration convened a council to advise the president "on the existing ability of FEMA to capably and impartially address disasters occurring within the United States" as well as to suggest changes to that process.[9] *NPR* obtained a copy of the council's draft report, which proposes raising the cost threshold for FEMA's recommendation formula, and/or basing declarations and assistance on the objective severity of the natural disaster itself instead of the damage done, e.g., the categorical severity of a hurricane/tornado/earthquake.[10] The final report has yet to come out, and whether its final recommendations will be heeded is unclear; complicated by leadership changes at the Department of Homeland Security.

But regardless of whether one believes that the Trump administration is just implementing more stringent standards for federal assistance in a nonpartisan way, it is clearly suboptimal for states to face such long wait times and such a high rejection rate for major disaster declaration requests. In the short term, they should make those standards clear so that local and state authorities do not waste time and effort on requests the Trump administration will ultimately reject. In the longer term, there are obvious ways to reform the disaster declaration process to make capricious presidential behavior less impactful, or even impossible altogether.

For instance, there are the regulatory 30-day time limits imposed on states for their initial request and for their appeal. Reciprocity here would go a long way. FEMA could develop a new rule giving themselves 30 days to issue their formal recommendation, but that would still not stop a president from sitting on that recommendation for as long as they liked. The more effective solution would be a congressional amendment to the Stafford Act, to add statutory deadlines to each part of the process: for instance, a 30-day deadline from the date of a request, or an appeal, for the president to act. This would curb the president's ability to punish states with long wait times. It would also put a ceiling on the rising wait time trend visible in the data even before the current presidency.

This would not, however, stop politically punitive declaration denials. *Politico*'s reporting, citing past FEMA and presidential officials, suggests that FEMA's calculations were historically the major, or even sole, basis for a president's decision.[11] If before this administration, presidential discretion was for practical purposes irrelevant, and during this administration, it is largely malign, why not do away with it altogether? Congress could go further than simply imposing statutory time limits and make it statutory law that specific calculable thresholds, which already contribute to FEMA's recommendation, determine whether states deserve federal funding for relief and not the fickle attitudes of the chief executive.

---

[9] "Federal Emergency Management Agency Review Council," DHS.Gov, https://www.dhs.gov/federal-emergency-management-agency-review-council.

[10] Lauren Sommer, "3 Big Changes Are Proposed for FEMA. This Is What Experts Really Think of Them," Climate, *NPR*, February 17, 2026, https://www.npr.org/2026/02/17/nx-s1-5698320/fema-reform-trump-hurricane-earthquake.

[11] Thomas Frank, "It's 3 Times Harder for Blue States to Get Disaster Funding under Trump," *POLITICO*, March 23, 2026, https://www.politico.com/news/2026/03/23/trump-denies-disaster-aid-for-democratic-led-states-00831199.

R Code For **All Plots**

```
# ------------------------------- SCRIPT INFO ---------------------------------
# fema_helpers.R - Shared helpers for FEMA disaster declaration analysis
#
# Contents:
#   §1 - Libraries
#   §2 - Reference data
#     §2a - Presidential admins (for Jan-20 year split + plot shading)
#     §2b - Rep/Dem state vote by presidential election (1976-2024)
#     §2c - Governor terms by state (1980-present, back-filled to Reagan)
#     §2d - Senator terms (loaded lazily from senator_terms.csv)
#   §3 - Load + clean FEMA data
#   §4 - Partisanship classifiers
#   §5 - Year bucketing (Jan-20 split)
#   §6 - Palette + method labels
#   §7 - Plot helpers (admin shading, president labels)
#   §8 - Incident type grouping
#   §9 - State populations + map colors
#
# Methodology dispatch:
#   classify_partisanship(disasters, method) where method is one of
#     "pres_vote" - state voted R or D in election of sitting president
#     "governor"  - party of sitting governor on decision date
#     "trifecta"  - majority of governor + 2 senators are same party
#
# To change colors: edit the PALETTE list in §6.
#
# -----------------------------------------------------------------------------

# ----------------------- §1: LIBRARIES ---------------------------------------

library(dplyr)
library(tidyverse)
library(ggtext)
library(extrafont)

# ----------------------- §2a: PRESIDENTIAL ADMINS ----------------------------

# One row per presidential term. election_year = the election that put them in.
# Used for (a) mapping decision date -> election year for pres_vote method,
# (b) Jan-20 year-split bucketing, (c) plot-side admin shading.
admins <- tribble(
  ~admin,             ~party, ~start,           ~end,             ~election_year,
  "Carter",           "D",    "1977-01-20",     "1981-01-20",     1976,
  "Reagan (1st)",     "R",    "1981-01-20",     "1985-01-20",     1980,
  "Reagan (2nd)",     "R",    "1985-01-20",     "1989-01-20",     1984,
  "Bush Sr.",         "R",    "1989-01-20",     "1993-01-20",     1988,
  "Clinton (1st)",    "D",    "1993-01-20",     "1997-01-20",     1992,
  "Clinton (2nd)",    "D",    "1997-01-20",     "2001-01-20",     1996,
  "Bush Jr. (1st)",   "R",    "2001-01-20",     "2005-01-20",     2000,
  "Bush Jr. (2nd)",   "R",    "2005-01-20",     "2009-01-20",     2004,
  "Obama (1st)",      "D",    "2009-01-20",     "2013-01-20",     2008,
  "Obama (2nd)",      "D",    "2013-01-20",     "2017-01-20",     2012,
  "Trump (1st)",      "R",    "2017-01-20",     "2021-01-20",     2016,
  "Biden",            "D",    "2021-01-20",     "2025-01-20",     2020,
  "Trump (2nd)",      "R",    "2025-01-20",     "2029-01-20",     2024
)
admins$start <- as.Date(admins$start)
admins$end   <- as.Date(admins$end)

# ----------------------- §2b: PRES VOTE BY ELECTION --------------------------

# States that voted REPUBLICAN in each election (1976-2024). States not listed
# voted Democratic. Source: MIT Election Data + Science Lab, cross-checked
# against the UCSB American Presidency Project.
```

```
rep_states <- list(
  "1976" = c("AK","AZ","CA","CO","CT","ID","IL","IN","IA","KS","ME","MI",
             "MT","NE","NV","NH","NJ","NM","ND","OK","OR","SD","UT","VA",
             "VT","WA","WY"),
  "1980" = c("AK","AL","AR","AZ","CA","CO","CT","DE","FL","ID","IL","IN",
             "IA","KS","KY","LA","ME","MA","MI","MS","MO","MT","NE","NV",
             "NH","NJ","NM","NY","NC","ND","OH","OK","OR","PA","SC","SD",
             "TN","TX","UT","VT","VA","WA","WI","WY"),
  "1984" = c("AK","AL","AR","AZ","CA","CO","CT","DE","FL","GA","HI","ID",
             "IL","IN","IA","KS","KY","LA","ME","MD","MA","MI","MS","MO",
             "MT","NE","NV","NH","NJ","NM","NY","NC","ND","OH","OK","OR",
             "PA","RI","SC","SD","TN","TX","UT","VT","VA","WA","WV","WI",
             "WY"),
  "1988" = c("AK","AL","AR","AZ","CA","CO","CT","DE","FL","GA","ID","IL",
             "IN","KS","KY","LA","ME","MD","MI","MS","MO","MT","NE","NV",
             "NH","NJ","NM","NC","ND","OH","OK","PA","SC","SD","TN","TX",
             "UT","VT","VA","WY"),
  "1992" = c("AL","AK","AZ","FL","ID","IN","KS","MS","NC","ND","NE","OK",
             "SC","SD","TX","UT","VA","WY"),
  "1996" = c("AL","AK","GA","ID","IN","KS","MS","MT","NC","ND","NE","OK",
             "SC","SD","TX","UT","VA","WY","CO"),
  "2000" = c("AK","AL","AR","AZ","CO","FL","GA","ID","IN","KS","KY","LA",
             "MS","MO","MT","NE","NV","NH","NC","ND","OH","OK","SC","SD",
             "TN","TX","UT","VA","WV","WY"),
  "2004" = c("AK","AL","AR","AZ","CO","FL","GA","ID","IN","IA","KS","KY",
             "LA","MS","MO","MT","NE","NV","NC","ND","NM","OH","OK","SC",
             "SD","TN","TX","UT","VA","WV","WY"),
  "2008" = c("AL","AK","AZ","AR","GA","ID","KS","KY","LA","MS","MO","MT",
             "NE","ND","OK","SC","SD","TN","TX","UT","WV","WY"),
  "2012" = c("AL","AK","AZ","AR","GA","ID","IN","KS","KY","LA","MS","MO",
             "MT","NE","NC","ND","OK","SC","SD","TN","TX","UT","WV","WY"),
  "2016" = c("AL","AK","AZ","AR","FL","GA","ID","IN","IA","KS","KY","LA",
             "MI","MS","MO","MT","NE","NC","ND","OH","OK","PA","SC","SD",
             "TN","TX","UT","WV","WI","WY"),
  "2020" = c("AL","AK","AR","FL","ID","IN","IA","KS","KY","LA","MS","MO",
             "MT","NE","NC","ND","OH","OK","SC","SD","TN","TX","UT","WV",
             "WY"),
  "2024" = c("AL","AK","AZ","AR","FL","GA","ID","IN","IA","KS","KY","LA",
             "MI","MS","MO","MT","NE","NV","NC","ND","OH","OK","PA","SC",
             "SD","TN","TX","UT","WV","WI","WY")
)

# State-level continuous Republican two-party vote share, averaged across
# 2016, 2020, 2024 elections. Generated by build_state_partisan_lean.R from
# tonmcg/US_County_Level_Election_Results_08-24.
state_partisan_lean <- NULL
state_partisan_lean_path <- "~/Documents/Classes/PPPA 6021 (Data Visualization)/Course
Data/state_partisan_lean.csv"
if (file.exists(state_partisan_lean_path)) {
  state_partisan_lean <- read.csv(state_partisan_lean_path)
}

# ---------------------- §2c: GOVERNOR TERMS (1980-present) -------------------

# Each row = the START of a period during which a given state was governed by a
# governor of the listed party. A row's party applies from its `start` date up
# to (but not including) the next row's `start` for that state.
#
# Parties:  "D" = Democratic, "R" = Republican, "I" = Independent/third-party
#           (Independent periods are excluded from the partisan analysis below.)
#
# Source: Ballotpedia
#

governor_terms <- tribble(
```

```
~state, ~start,         ~party,
# Alabama
"AL",   "1979-01-15",   "D",   # Fob James -> George Wallace
"AL",   "1987-01-19",   "R",   # Guy Hunt (in office through 1993 removal)
"AL",   "1993-04-22",   "D",   # Jim Folsom Jr. (Hunt removed)
"AL",   "1995-01-16",   "R",   # Fob James (second run, now R)
"AL",   "1999-01-18",   "D",   # Don Siegelman
"AL",   "2003-01-20",   "R",   # Bob Riley -> Bentley -> Ivey
# Alaska
"AK",   "1974-12-02",   "R",   # Jay Hammond (in office at window start)
"AK",   "1982-12-06",   "D",   # Bill Sheffield -> Steve Cowper
"AK",   "1990-12-03",   "I",   # Wally Hickel (AIP)
"AK",   "1994-12-05",   "D",   # Tony Knowles
"AK",   "2002-12-02",   "R",   # Murkowski -> Palin -> Parnell
"AK",   "2014-12-01",   "I",   # Bill Walker
"AK",   "2018-12-03",   "R",   # Mike Dunleavy
# Arizona
"AZ",   "1978-03-04",   "D",   # Bruce Babbitt (in office at window start)
"AZ",   "1987-01-05",   "R",   # Evan Mecham
"AZ",   "1988-04-04",   "D",   # Rose Mofford (Mecham impeached)
"AZ",   "1991-03-06",   "R",   # Symington -> Hull
"AZ",   "2003-01-06",   "D",   # Janet Napolitano
"AZ",   "2009-01-21",   "R",   # Brewer -> Ducey
"AZ",   "2023-01-02",   "D",   # Katie Hobbs
# Arkansas
"AR",   "1979-01-09",   "D",   # Bill Clinton (1st term)
"AR",   "1981-01-19",   "R",   # Frank White
"AR",   "1983-01-11",   "D",   # Clinton (again) -> Tucker
"AR",   "1996-07-15",   "R",   # Mike Huckabee
"AR",   "2007-01-09",   "D",   # Mike Beebe
"AR",   "2015-01-13",   "R",   # Hutchinson -> Sanders
# California
"CA",   "1975-01-06",   "D",   # Jerry Brown (1st stint, in office at start)
"CA",   "1983-01-03",   "R",   # Deukmejian -> Wilson
"CA",   "1999-01-04",   "D",   # Gray Davis
"CA",   "2003-11-17",   "R",   # Schwarzenegger (recall)
"CA",   "2011-01-03",   "D",   # Brown -> Newsom
# Colorado
"CO",   "1975-01-14",   "D",   # Richard Lamm -> Roy Romer
"CO",   "1999-01-12",   "R",   # Bill Owens
"CO",   "2007-01-09",   "D",   # Ritter -> Hickenlooper -> Polis
# Connecticut
"CT",   "1975-01-08",   "D",   # Ella Grasso -> William O'Neill
"CT",   "1991-01-09",   "I",   # Lowell Weicker (A Connecticut Party)
"CT",   "1995-01-04",   "R",   # Rowland -> Rell
"CT",   "2011-01-05",   "D",   # Malloy -> Lamont
# Delaware
"DE",   "1977-01-18",   "R",   # Pete du Pont -> Michael Castle
"DE",   "1993-01-19",   "D",   # Carper -> Minner -> Markell -> Carney -> Meyer
# District of Columbia
"DC",   "1979-01-02",   "D",   # Marion Barry onward — continuous D mayors
# Florida
"FL",   "1979-01-02",   "D",   # Bob Graham
"FL",   "1987-01-06",   "R",   # Bob Martinez
"FL",   "1991-01-08",   "D",   # Chiles -> MacKay
"FL",   "1999-01-05",   "R",   # Jeb Bush -> Crist -> Scott -> DeSantis
# Georgia
"GA",   "1975-01-14",   "D",   # Busbee -> Harris -> Miller -> Barnes
"GA",   "2003-01-13",   "R",   # Perdue -> Deal -> Kemp
# Hawaii
"HI",   "1974-12-02",   "D",   # Ariyoshi -> Waihee -> Cayetano
"HI",   "2002-12-02",   "R",   # Linda Lingle
"HI",   "2010-12-06",   "D",   # Abercrombie -> Ige -> Green
# Idaho
"ID",   "1977-01-24",   "D",   # John Evans -> Cecil Andrus
```

```
"ID",   "1995-01-02",   "R",   # Batt -> Kempthorne -> Risch -> Otter -> Little
# Illinois
"IL",   "1977-01-10",   "R",   # Thompson -> Edgar -> Ryan
"IL",   "2003-01-13",   "D",   # Blagojevich -> Quinn
"IL",   "2015-01-12",   "R",   # Bruce Rauner
"IL",   "2019-01-14",   "D",   # J.B. Pritzker
# Indiana
"IN",   "1973-01-08",   "R",   # Bowen -> Orr
"IN",   "1989-01-09",   "D",   # Bayh -> O'Bannon -> Kernan
"IN",   "2005-01-10",   "R",   # Daniels -> Pence -> Holcomb -> Braun
# Iowa
"IA",   "1969-01-16",   "R",   # Ray -> Branstad (1st tenure)
"IA",   "1999-01-15",   "D",   # Vilsack -> Culver
"IA",   "2011-01-14",   "R",   # Branstad (2nd) -> Reynolds
# Kansas
"KS",   "1979-01-08",   "D",   # John Carlin
"KS",   "1987-01-12",   "R",   # Mike Hayden
"KS",   "1991-01-14",   "D",   # Joan Finney
"KS",   "1995-01-09",   "R",   # Bill Graves
"KS",   "2003-01-13",   "D",   # Sebelius -> Parkinson
"KS",   "2011-01-10",   "R",   # Brownback -> Colyer
"KS",   "2019-01-14",   "D",   # Laura Kelly
# Kentucky
"KY",   "1979-12-11",   "D",   # Brown -> Collins -> Wilkinson -> Jones -> Patton
"KY",   "2003-12-09",   "R",   # Ernie Fletcher
"KY",   "2007-12-11",   "D",   # Steve Beshear
"KY",   "2015-12-08",   "R",   # Matt Bevin
"KY",   "2019-12-10",   "D",   # Andy Beshear
# Louisiana
"LA",   "1972-05-09",   "D",   # Edwin Edwards (1st two terms, in office at start)
"LA",   "1980-03-10",   "R",   # David Treen (first R LA gov since Reconstruction)
"LA",   "1984-03-12",   "D",   # Edwards (3rd term) -> Roemer (as D)
"LA",   "1991-03-11",   "R",   # Roemer switched D -> R
"LA",   "1992-01-13",   "D",   # Edwin Edwards (4th term)
"LA",   "1996-01-08",   "R",   # Mike Foster
"LA",   "2004-01-12",   "D",   # Kathleen Blanco
"LA",   "2008-01-14",   "R",   # Bobby Jindal
"LA",   "2016-01-11",   "D",   # John Bel Edwards
"LA",   "2024-01-08",   "R",   # Jeff Landry
# Maine
"ME",   "1979-01-03",   "D",   # Joseph Brennan
"ME",   "1987-01-07",   "R",   # John McKernan
"ME",   "1995-01-05",   "I",   # Angus King
"ME",   "2003-01-08",   "D",   # John Baldacci
"ME",   "2011-01-05",   "R",   # Paul LePage
"ME",   "2019-01-02",   "D",   # Janet Mills
# Maryland
"MD",   "1979-01-17",   "D",   # Hughes -> Schaefer -> Glendening
"MD",   "2003-01-15",   "R",   # Bob Ehrlich
"MD",   "2007-01-17",   "D",   # Martin O'Malley
"MD",   "2015-01-21",   "R",   # Larry Hogan
"MD",   "2023-01-18",   "D",   # Wes Moore
# Massachusetts
"MA",   "1979-01-04",   "D",   # Edward King -> Michael Dukakis
"MA",   "1991-01-03",   "R",   # Weld -> Cellucci -> Swift -> Romney
"MA",   "2007-01-04",   "D",   # Deval Patrick
"MA",   "2015-01-08",   "R",   # Charlie Baker
"MA",   "2023-01-05",   "D",   # Maura Healey
# Michigan
"MI",   "1969-01-22",   "R",   # William Milliken (in office at window start)
"MI",   "1983-01-01",   "D",   # James Blanchard
"MI",   "1991-01-01",   "R",   # John Engler
"MI",   "2003-01-01",   "D",   # Jennifer Granholm
"MI",   "2011-01-01",   "R",   # Rick Snyder
"MI",   "2019-01-01",   "D",   # Gretchen Whitmer
```

```
# Minnesota
"MN",   "1979-01-04",   "R",   # Albert Quie
"MN",   "1983-01-03",   "D",   # Rudy Perpich (DFL)
"MN",   "1991-01-07",   "R",   # Arne Carlson
"MN",   "1999-01-04",   "I",   # Jesse Ventura (Reform)
"MN",   "2003-01-06",   "R",   # Tim Pawlenty
"MN",   "2011-01-03",   "D",   # Dayton -> Walz (DFL)
# Mississippi
"MS",   "1976-01-20",   "D",   # Finch -> Winter -> Allain -> Mabus
"MS",   "1992-01-14",   "R",   # Kirk Fordice
"MS",   "2000-01-11",   "D",   # Ronnie Musgrove
"MS",   "2004-01-13",   "R",   # Barbour -> Bryant -> Reeves
# Missouri
"MO",   "1977-01-10",   "D",   # Joseph Teasdale
"MO",   "1981-01-12",   "R",   # Bond -> Ashcroft
"MO",   "1993-01-11",   "D",   # Carnahan -> Wilson -> Holden
"MO",   "2005-01-10",   "R",   # Matt Blunt
"MO",   "2009-01-12",   "D",   # Jay Nixon
"MO",   "2017-01-09",   "R",   # Greitens -> Parson -> Kehoe
# Montana
"MT",   "1973-01-06",   "D",   # Judge -> Schwinden
"MT",   "1989-01-02",   "R",   # Stephens -> Racicot -> Martz
"MT",   "2005-01-03",   "D",   # Schweitzer -> Bullock
"MT",   "2021-01-04",   "R",   # Greg Gianforte
# Nebraska
"NE",   "1979-01-04",   "R",   # Charles Thone
"NE",   "1983-01-06",   "D",   # Bob Kerrey
"NE",   "1987-01-09",   "R",   # Kay Orr
"NE",   "1991-01-09",   "D",   # Ben Nelson
"NE",   "1999-01-07",   "R",   # Johanns -> Heineman -> Ricketts -> Pillen
# Nevada
"NV",   "1979-01-01",   "R",   # Robert List
"NV",   "1983-01-03",   "D",   # Richard Bryan -> Bob Miller
"NV",   "1999-01-04",   "R",   # Guinn -> Gibbons -> Sandoval
"NV",   "2019-01-07",   "D",   # Steve Sisolak
"NV",   "2023-01-02",   "R",   # Joe Lombardo
# New Hampshire
"NH",   "1979-01-04",   "D",   # Hugh Gallen
"NH",   "1983-01-06",   "R",   # Sununu (J.) -> Gregg -> Merrill
"NH",   "1997-01-09",   "D",   # Jeanne Shaheen
"NH",   "2003-01-09",   "R",   # Craig Benson
"NH",   "2005-01-06",   "D",   # Lynch -> Hassan
"NH",   "2017-01-05",   "R",   # Sununu (C.) -> Ayotte
# New Jersey
"NJ",   "1974-01-15",   "D",   # Brendan Byrne
"NJ",   "1982-01-19",   "R",   # Tom Kean
"NJ",   "1990-01-16",   "D",   # Jim Florio
"NJ",   "1994-01-18",   "R",   # Whitman -> DiFrancesco
"NJ",   "2002-01-15",   "D",   # McGreevey -> Codey -> Corzine
"NJ",   "2010-01-19",   "R",   # Chris Christie
"NJ",   "2018-01-16",   "D",   # Murphy -> Sherrill
# New Mexico
"NM",   "1979-01-01",   "D",   # Bruce King (2nd term) -> Toney Anaya
"NM",   "1987-01-01",   "R",   # Garrey Carruthers
"NM",   "1991-01-01",   "D",   # Bruce King (3rd term)
"NM",   "1995-01-01",   "R",   # Gary Johnson
"NM",   "2003-01-01",   "D",   # Bill Richardson
"NM",   "2011-01-01",   "R",   # Susana Martinez
"NM",   "2019-01-01",   "D",   # Michelle Lujan Grisham
# New York
"NY",   "1975-01-01",   "D",   # Hugh Carey -> Mario Cuomo
"NY",   "1995-01-01",   "R",   # George Pataki
"NY",   "2007-01-01",   "D",   # Spitzer -> Paterson -> Cuomo (A.) -> Hochul
# North Carolina
"NC",   "1977-01-08",   "D",   # Jim Hunt (first two terms)
```

```
  "NC",   "1985-01-05",   "R",   # Jim Martin
  "NC",   "1993-01-09",   "D",   # Hunt -> Easley -> Perdue
  "NC",   "2013-01-05",   "R",   # Pat McCrory
  "NC",   "2017-01-01",   "D",   # Cooper -> Stein
  # North Dakota
  "ND",   "1973-01-01",   "D",   # Art Link
  "ND",   "1981-01-06",   "R",   # Allen Olson
  "ND",   "1985-01-01",   "D",   # George Sinner
  "ND",   "1992-12-15",   "R",   # Schafer -> Hoeven -> Dalrymple -> Burgum -> Armstrong
  # Ohio
  "OH",   "1975-01-13",   "R",   # James Rhodes (2nd two terms)
  "OH",   "1983-01-10",   "D",   # Dick Celeste
  "OH",   "1991-01-14",   "R",   # Voinovich -> Taft
  "OH",   "2007-01-08",   "D",   # Ted Strickland
  "OH",   "2011-01-10",   "R",   # Kasich -> DeWine
  # Oklahoma
  "OK",   "1979-01-13",   "D",   # George Nigh
  "OK",   "1987-01-12",   "R",   # Henry Bellmon
  "OK",   "1991-01-14",   "D",   # David Walters
  "OK",   "1995-01-09",   "R",   # Frank Keating
  "OK",   "2003-01-13",   "D",   # Brad Henry
  "OK",   "2011-01-10",   "R",   # Fallin -> Stitt
  # Oregon
  "OR",   "1979-01-08",   "R",   # Victor Atiyeh
  "OR",   "1987-01-12",   "D",   # Goldschmidt -> Roberts -> Kitzhaber -> Kulongoski ->
Brown -> Kotek
  # Pennsylvania
  "PA",   "1979-01-16",   "R",   # Dick Thornburgh
  "PA",   "1987-01-20",   "D",   # Bob Casey Sr.
  "PA",   "1995-01-17",   "R",   # Ridge -> Schweiker
  "PA",   "2003-01-21",   "D",   # Ed Rendell
  "PA",   "2011-01-18",   "R",   # Tom Corbett
  "PA",   "2015-01-20",   "D",   # Wolf -> Shapiro
  # Rhode Island
  "RI",   "1977-01-04",   "D",   # J. Joseph Garrahy
  "RI",   "1985-01-01",   "R",   # Edward DiPrete
  "RI",   "1991-01-01",   "D",   # Bruce Sundlun
  "RI",   "1995-01-03",   "R",   # Almond -> Carcieri
  "RI",   "2011-01-04",   "I",   # Lincoln Chafee (as Independent)
  "RI",   "2013-05-29",   "D",   # Chafee switched I -> D; then Raimondo -> McKee
  # South Carolina
  "SC",   "1979-01-10",   "D",   # Richard Riley
  "SC",   "1987-01-14",   "R",   # Campbell -> Beasley
  "SC",   "1999-01-13",   "D",   # Jim Hodges
  "SC",   "2003-01-15",   "R",   # Sanford -> Haley -> McMaster
  # South Dakota
  "SD",   "1979-01-01",   "R",   # Janklow -> Mickelson -> Miller -> Janklow -> Rounds ->
Daugaard -> Noem
  # Tennessee
  "TN",   "1979-01-17",   "R",   # Lamar Alexander
  "TN",   "1987-01-17",   "D",   # Ned McWherter
  "TN",   "1995-01-21",   "R",   # Don Sundquist
  "TN",   "2003-01-18",   "D",   # Phil Bredesen
  "TN",   "2011-01-15",   "R",   # Haslam -> Lee
  # Texas
  "TX",   "1979-01-16",   "R",   # Bill Clements (1st term)
  "TX",   "1983-01-18",   "D",   # Mark White
  "TX",   "1987-01-20",   "R",   # Clements (2nd term)
  "TX",   "1991-01-15",   "D",   # Ann Richards
  "TX",   "1995-01-17",   "R",   # Bush (G.W.) -> Perry -> Abbott
  # Utah
  "UT",   "1977-01-03",   "D",   # Scott Matheson
  "UT",   "1985-01-07",   "R",   # Bangerter -> Leavitt -> Walker -> Huntsman -> Herbert -
> Cox
  # Vermont
```

```
  "VT",   "1977-01-04",   "R",   # Richard Snelling (1st multi-term stint)
  "VT",   "1985-01-10",   "D",   # Madeleine Kunin
  "VT",   "1991-01-10",   "R",   # Richard Snelling (2nd stint)
  "VT",   "1991-08-14",   "D",   # Howard Dean (Snelling died)
  "VT",   "2003-01-09",   "R",   # Jim Douglas
  "VT",   "2011-01-06",   "D",   # Peter Shumlin
  "VT",   "2017-01-05",   "R",   # Phil Scott
  # Virginia
  "VA",   "1978-01-14",   "R",   # John Dalton
  "VA",   "1982-01-16",   "D",   # Robb -> Baliles -> Wilder
  "VA",   "1994-01-15",   "R",   # Allen -> Gilmore
  "VA",   "2002-01-12",   "D",   # Warner -> Kaine
  "VA",   "2010-01-16",   "R",   # Bob McDonnell
  "VA",   "2014-01-11",   "D",   # McAuliffe -> Northam
  "VA",   "2022-01-15",   "R",   # Glenn Youngkin
  "VA",   "2026-01-17",   "D",   # Abigail Spanberger
  # Washington
  "WA",   "1977-01-12",   "D",   # Dixy Lee Ray
  "WA",   "1981-01-14",   "R",   # John Spellman
  "WA",   "1985-01-16",   "D",   # Gardner -> Lowry -> Locke -> Gregoire -> Inslee ->
Ferguson
  # West Virginia
  "WV",   "1977-01-17",   "D",   # Jay Rockefeller
  "WV",   "1985-01-14",   "R",   # Arch Moore
  "WV",   "1989-01-16",   "D",   # Caperton
  "WV",   "1997-01-13",   "R",   # Cecil Underwood
  "WV",   "2001-01-15",   "D",   # Wise -> Manchin -> Tomblin -> Justice (as Dem)
  "WV",   "2017-08-03",   "R",   # Jim Justice switched D -> R; then Morrisey
  # Wisconsin
  "WI",   "1979-01-01",   "R",   # Lee Dreyfus
  "WI",   "1983-01-03",   "D",   # Tony Earl
  "WI",   "1987-01-05",   "R",   # Thompson -> McCallum
  "WI",   "2003-01-06",   "D",   # Jim Doyle
  "WI",   "2011-01-03",   "R",   # Scott Walker
  "WI",   "2019-01-07",   "D",   # Tony Evers
  # Wyoming
  "WY",   "1975-01-06",   "D",   # Ed Herschler -> Mike Sullivan
  "WY",   "1995-01-02",   "R",   # Jim Geringer
  "WY",   "2003-01-06",   "D",   # Dave Freudenthal
  "WY",   "2011-01-03",   "R",   # Mead -> Gordon
)
governor_terms$start <- as.Date(governor_terms$start)
governor_terms <- governor_terms[order(governor_terms$state, governor_terms$start), ]

# ---------------------- §2d: SENATOR TERMS (lazy load) ----------------------

# Generated by build_senator_terms.R from the @unitedstates/congress-legislators
# project. Columns: state, class, start, end, party ("D"/"R"; independents
# mapped to their caucus party).
#
# Only load if the file exists; the trifecta classifier will error with a
# helpful message if senator_terms is needed but not loaded.
senator_terms <- NULL
senator_terms_path <- "~/Documents/Classes/PPPA 6021 (Data Visualization)/Course
Data/senator_terms.csv"
if (file.exists(senator_terms_path)) {
  senator_terms <- read.csv(senator_terms_path)
  senator_terms$start <- as.Date(senator_terms$start)
  senator_terms$end   <- as.Date(senator_terms$end)
}

# ---------------------- §3: LOAD + CLEAN FEMA DATA -------------------------

# Load + outer-join the FEMA declarations and denials data sets.
load_fema <- function(decl_path, den_path) {
```

```
  declarations <- read.csv(decl_path)
  denials      <- read.csv(den_path)

  # Slim to only the columns needed for the merge and analysis
  declarations <- declarations[, c("disasterNumber", "declarationDate",
                                   "declarationType", "stateCode", "stateName",
                                   "incidentType", "incidentBeginDate",
                                   "incidentEndDate", "declarationRequestDate")]
  denials <- denials[, c("declarationRequestNumber", "stateAbbreviation", "state",
                         "declarationRequestType", "declarationRequestDate",
                         "requestedIncidentTypes", "requestedIncidentBeginDate",
                         "requestedIncidentEndDate", "currentRequestStatus",
                         "requestStatusDate")]

  # Full outer join: approvals + denials, matched where possible.
  # After merge: denial's requestStatusDate ends up in declarationDate column.
  disasters <- merge(
    x = declarations,
    y = denials,
    by.x = c("disasterNumber", "stateCode", "stateName", "declarationType",
             "incidentType", "incidentBeginDate", "incidentEndDate",
             "declarationRequestDate", "declarationDate"),
    by.y = c("declarationRequestNumber", "stateAbbreviation", "state",
             "declarationRequestType", "requestedIncidentTypes",
             "requestedIncidentBeginDate", "requestedIncidentEndDate",
             "declarationRequestDate", "requestStatusDate"),
    all = TRUE
  )

  return(disasters)
}

# Clean + filter the merged data: trim, recode, parse dates, fix typos, etc.
clean_disasters <- function(disasters) {
  # Trim whitespace on state fields
  disasters$stateName <- trimws(disasters$stateName)
  disasters$stateCode <- trimws(disasters$stateCode)

  # Recode request status: unmatched declarations = Approved, Turndown = Denied
  disasters$currentRequestStatus[disasters$currentRequestStatus == "Turndown"] <- "Denied"
  disasters$currentRequestStatus[is.na(disasters$currentRequestStatus)] <- "Approved"

  # Parse date strings -> Date objects (strip trailing time component)
  for (col in c("incidentBeginDate", "incidentEndDate",
                "declarationRequestDate", "declarationDate")) {
    disasters[[col]] <- substr(disasters[[col]], 1, nchar(disasters[[col]]) - 14)
    disasters[[col]] <- as.Date(disasters[[col]])
  }

  # Fix known date typos in the source data
  disasters$incidentBeginDate[disasters$incidentBeginDate == as.Date("0003-05-10")] <-
as.Date("2003-05-10")
  disasters$declarationRequestDate[disasters$declarationRequestDate == as.Date("0999-08-
05")] <- as.Date("1999-08-05")
  disasters$incidentBeginDate[disasters$disasterNumber == 1107 &
                                disasters$stateCode == "OR"] <- as.Date("1995-12-10")
  disasters$incidentEndDate[disasters$disasterNumber == 1107 &
                              disasters$stateCode == "OR"] <- as.Date("1995-12-12")

  # Keep only Major Disaster declarations (drops Emergency + Fire Mgmt)
  disasters <- filter(disasters, declarationType == "Major Disaster")

  # Drop freely associated states
  disasters <- filter_out(disasters, stateCode %in% c("FM", "MH", "PW"))
```

```
  # Drop U.S. territories
  disasters <- filter_out(disasters, stateCode %in% c("PR", "VI", "GU", "AS", "MP"))

  # Drop COVID biological declarations (all share the same 2020-01-20 start)
  disasters <- filter_out(disasters, incidentType == "Biological" &
                            incidentBeginDate == as.Date("2020-01-20"))

  # Unified decision date (approvals + denials both land in declarationDate)
  disasters$decisionDate <- disasters$declarationDate

  return(disasters)
}

# ---------------------- §4a: pres_vote CLASSIFIER ---------------------------

# Tag each row with state_party based on how the state voted in the election
# of the sitting president at the time of the decision.
classify_pres_vote <- function(d) {
  # Tag each row with the election year of the president who decided
  d$election_year <- NA_integer_
  for (i in 1:nrow(admins)) {
    mask <- !is.na(d$decisionDate) &
      d$decisionDate >= admins$start[i] &
      d$decisionDate <  admins$end[i]
    d$election_year[mask] <- admins$election_year[i]
  }

  # Label each state as Republican or Democratic based on that election
  d$state_party <- NA_character_
  for (yr in names(rep_states)) {
    yr_num <- as.integer(yr)
    mask <- !is.na(d$election_year) & d$election_year == yr_num
    d$state_party[mask] <- ifelse(
      d$stateCode[mask] %in% rep_states[[yr]],
      "Republican",
      "Democratic"
    )
  }

  # Drop unclassifiable rows (pre-Carter or missing decision date)
  d <- filter(d, !is.na(state_party))
  return(d)
}

# ---------------------- §4b: governor CLASSIFIER ----------------------------

# Helper: look up governor party ("D"/"R"/"I"/NA) for each row, without
# filtering. Used by both classify_governor() and classify_trifecta().
governor_party_on_decision <- function(d) {
  out <- rep(NA_character_, nrow(d))
  for (s in unique(governor_terms$state)) {
    rows <- which(d$stateCode == s & !is.na(d$decisionDate))
    if (length(rows) == 0) next
    terms_s <- governor_terms[governor_terms$state == s, ]   # pre-sorted
    # findInterval: terms_s$start[i] <= date < terms_s$start[i+1]
    idx <- findInterval(d$decisionDate[rows], terms_s$start)
    idx[idx == 0] <- NA
    out[rows] <- terms_s$party[idx]
  }
  return(out)
}

# Classifier: add state_party ("Democratic"/"Republican"), drop Independent
# periods and unclassifiable rows.
classify_governor <- function(d) {
```

```
  gov <- governor_party_on_decision(d)
  d$state_party <- case_when(
    gov == "R" ~ "Republican",
    gov == "D" ~ "Democratic",
    TRUE       ~ NA_character_     # "I" and NA both drop
  )
  d <- filter(d, !is.na(state_party))
  return(d)
}

# ---------------------- §4c: trifecta CLASSIFIER ----------------------------

# Helper: for each row, return a 2-column character matrix giving the parties
# ("D"/"R") of the two senate seats for that state on the decision date.
# Independents are already mapped to caucus party in senator_terms.csv.
# Rows where either seat can't be resolved get NA.
#
# Works per Senate class (each state has 2 of classes {1,2,3}), using
# findInterval so term-boundary days behave correctly.
senator_parties_on_decision <- function(d) {
  out <- matrix(NA_character_, nrow = nrow(d), ncol = 2)
  for (s in unique(d$stateCode)) {
    rows <- which(d$stateCode == s & !is.na(d$decisionDate))
    if (length(rows) == 0) next

    terms_s <- senator_terms[senator_terms$state == s, ]
    if (nrow(terms_s) == 0) next   # no data for this state

    classes_s <- sort(unique(terms_s$class))   # usually length 2
    for (i in seq_along(classes_s)) {
      if (i > 2) break   # defensive: a state should have exactly 2 classes
      terms_cls <- terms_s[terms_s$class == classes_s[i], ]
      terms_cls <- terms_cls[order(terms_cls$start), ]

      # Latest term whose start <= dd; NA if pre-first-term
      idx <- findInterval(d$decisionDate[rows], terms_cls$start)
      idx[idx == 0] <- NA

      # Reject if dd is past that term's end (e.g., a resignation gap)
      matched_party <- terms_cls$party[idx]
      matched_end   <- terms_cls$end[idx]
      in_term <- !is.na(idx) & d$decisionDate[rows] <= matched_end
      matched_party[!in_term] <- NA

      out[rows, i] <- matched_party
    }
  }
  return(out)
}

# Classifier: count parties among (governor, senator 1, senator 2). State
# is "Democratic" if more D than R, "Republican" if more R than D, dropped
# if tied or any senator data is missing. Independent governors contribute
# 0 to both counts (they're effectively a "wildcard" — both senators must
# agree to break the tie).
classify_trifecta <- function(d) {
  if (is.null(senator_terms)) {
    stop("senator_terms not loaded. Run build_senator_terms.R to generate",
         " senator_terms.csv in the working directory.")
  }

  gov  <- governor_party_on_decision(d)           # "D"/"R"/"I"/NA
  sens <- senator_parties_on_decision(d)          # Nx2 matrix "D"/"R"/NA

  # Drop rows with any missing data (gov NA or either senator NA)
```

```
  missing <- is.na(gov) | is.na(sens[, 1]) | is.na(sens[, 2])

  # Count D and R across the three actors. I-governor contributes 0 to both.
  # is.na(gov) is already in `missing` so safe to compare.
  gov_safe <- ifelse(is.na(gov), "X", gov)
  d_count <- as.integer(gov_safe == "D") +
    as.integer(sens[, 1] == "D") +
    as.integer(sens[, 2] == "D")
  r_count <- as.integer(gov_safe == "R") +
    as.integer(sens[, 1] == "R") +
    as.integer(sens[, 2] == "R")

  d$state_party <- case_when(
    !missing & d_count > r_count ~ "Democratic",
    !missing & r_count > d_count ~ "Republican",
    TRUE                         ~ NA_character_   # tied or missing -> drop
  )

  d <- filter(d, !is.na(state_party))
  return(d)
}

# ---------------------- §4d: DISPATCHER -------------------------------------

# Pick a classifier by method name. Plot scripts call this directly.
classify_partisanship <- function(d, method) {
  if (method == "pres_vote") return(classify_pres_vote(d))
  if (method == "governor")  return(classify_governor(d))
  if (method == "trifecta")  return(classify_trifecta(d))
  stop("Unknown method: ", method,
       " (expected 'pres_vote', 'governor', or 'trifecta')")
}

# ---------------------- §5: YEAR BUCKETING (Jan-20 split) -------------------

# Bucket decision dates into "presidential year" (year boundary = Jan 20, not
# Jan 1). E.g., decision on Jan 5 2025 -> 2024 (Biden's last year); decision
# on Jan 25 2025 -> 2025 (Trump's first year). Accepts a Date vector, returns
# an integer vector of the same length.
compute_presidential_year <- function(dates) {
  cal_year <- as.numeric(format(dates, "%Y"))
  jan20    <- as.Date(paste0(cal_year, "-01-20"))
  passed   <- dates >= jan20
  year     <- ifelse(passed, cal_year, cal_year - 1)
  return(year)
}

# ---------------------- §6: PALETTE + METHOD LABELS -------------------------

# Central color palette
PALETTE <- list(
  Democratic = "#2471A3",
  Republican = "#C0392B",
  Mixed      = "#7D3C98" # not used right now
)

# Return the color vector for a given method. Names must match state_party
# values so scale_color_manual(values = get_palette(method)) works.
# All three methods now return only Democratic + Republican.
get_palette <- function(method) {
  return(c(
    Democratic = PALETTE$Democratic,
    Republican = PALETTE$Republican
  ))
}
```

```
# Method-specific wording for subtitles / captions in plot scripts.
get_method_labels <- function(method) {
  if (method == "pres_vote") {
    return(list(
      d_phrase     = "Democratic-voting states",
      r_phrase     = "Republican-voting states",
      mixed_phrase = NULL,
      method_note  = paste0("State partisanship is based on the state's ",
                            "vote in the election for the sitting president.")
    ))
  }
  if (method == "governor") {
    return(list(
      d_phrase     = "Democratic-governed states",
      r_phrase     = "Republican-governed states",
      mixed_phrase = NULL,
      method_note  = paste0("State partisanship is based on the party of ",
                            "the sitting governor on the decision date.")
    ))
  }
  if (method == "trifecta") {
    return(list(
      d_phrase     = "Democratic-controlled states",
      r_phrase     = "Republican-controlled states",
      mixed_phrase = NULL,
      method_note  = paste0("State partisanship is based on which party ",
                            "holds more of the governor + senate seats.")
    ))
  }
  stop("Unknown method: ", method)
}

# ---------------------- §8: INCIDENT TYPE GROUPING -------------------------

# FEMA records 25+ distinct incident types, many with very few cases. Group
# into 7 readable buckets for stacked/area-style visualizations. Order of
# the factor levels controls stack order in plots (bottom -> top).
INCIDENT_GROUPS <- c(
  "Severe Storm"          = "Severe Storm",
  "Flood"                 = "Flood",
  "Hurricane"             = "Tropical Cyclone",
  "Typhoon"               = "Tropical Cyclone",
  "Tropical Storm"        = "Tropical Cyclone",
  "Tropical Depression"   = "Tropical Cyclone",
  "Coastal Storm"         = "Severe Storm",
  "Tornado"               = "Severe Storm",
  "Snowstorm"             = "Winter Weather",
  "Severe Ice Storm"      = "Winter Weather",
  "Winter Storm"          = "Winter Weather",
  "Freezing"              = "Winter Weather",
  "Fire"                  = "Fire",
  "Earthquake"            = "Geophysical",
  "Volcanic Eruption"     = "Geophysical",
  "Tsunami"               = "Geophysical",
  "Mud/Landslide"         = "Geophysical",
  "Dam/Levee Break"       = "Geophysical",
  "Straight-Line Winds"   = "Severe Storm"
)

# Stack/legend order: most-common buckets at the bottom, rare at the top
INCIDENT_LEVELS <- c(
  "Severe Storm", "Flood", "Tropical Cyclone",
  "Winter Weather", "Fire", "Geophysical", "Other"
)
```

```
# Color palette.
INCIDENT_COLORS <- c(
  "Severe Storm"     = "#f7aa58",
  "Flood"            = "#376795",
  "Tropical Cyclone" = "#b695bc",
  "Winter Weather"   = "#aadce0",
  "Fire"             = "#e76254",
  "Geophysical"      = "#6f9969",
  "Other"            = "grey60"
)

# Map incidentType -> grouped category factor with fixed level order.
group_incident_type <- function(incidentType) {
  grouped <- INCIDENT_GROUPS[as.character(incidentType)]
  grouped[is.na(grouped)] <- "Other"
  factor(grouped, levels = INCIDENT_LEVELS)
}

# ---------------------- §9: STATE POPULATIONS + MAP COLORS -----------------

# State population data
#
# Source: U.S. Census Bureau, Population Estimates Program (PEP).
#         https://www.census.gov/programs-surveys/popest.html
#
state_population <- NULL
state_population_path <- "~/Documents/Classes/PPPA 6021 (Data Visualization)/Course
Data/state_population.csv"
if (file.exists(state_population_path)) {
  state_population <- read.csv(state_population_path)
}

# Helper: average population for a state over a window (inclusive year range).
# Returns a data frame with columns state, avg_population. If state_population
# isn't loaded, errors with a helpful message pointing to the builder.
state_pop_window_mean <- function(year_min, year_max) {
  if (is.null(state_population)) {
    stop("state_population not loaded. Run build_state_population.R to ",
         "generate state_population.csv.")
  }
  win <- filter(state_population,
                year >= year_min, year <= year_max)
  if (nrow(win) == 0) {
    stop("No state_population rows in window ", year_min, "-", year_max,
         ". Available years: ", min(state_population$year), "-",
         max(state_population$year))
  }
  out <- summarise(group_by(win, state),
                   avg_population = mean(population),
                   .groups = "drop")
  return(out)
}

# Choropleth color palettes (NOT USED RIGHT NOW)
# MAP_PALETTE <- list(
#   # Sequential (low -> high)
#   seq_low   = "#F7F4F0",
#   seq_high  = "#7B3F00",
#   # Diverging (negative -> zero -> positive)
#   div_neg   = "#5E3C99",   # purple
#   div_mid   = "#F7F4F0",   # off-white
#   div_pos   = "#E66101"    # orange
# )
```

```r
# ---------------------- §7: PLOT HELPERS (shading + labels) ------------------

# Convention: data points are positioned at integer presidential-years (1989,
# 1990, ...), so a year-bucket is centered on its integer. Shading regions and
# president-label brackets use offsets of 0.5 around those integers so each
# bucket is fully covered.

# Admin shading rectangles clipped to the plot window. Each row is one admin
# term; returns admin, party, year_start, year_end. Use as xmin/xmax for
# annotate("rect"). Pass integer year_min/year_max (e.g. 1989, 2026).
build_admin_shading <- function(year_min, year_max) {
  left  <- year_min - 0.5
  right <- year_max + 0.5

  shade <- admins
  shade$year_start <- pmax(as.numeric(format(admins$start, "%Y")) - 0.5, left)
  shade$year_end   <- pmin(as.numeric(format(admins$end,   "%Y")) - 0.5, right)
  shade <- filter(shade, year_end > left, year_start < right)
  shade <- shade[, c("admin", "party", "year_start", "year_end")]
  return(shade)
}

# President-name labels for the bracket above the plot area. Merges
# consecutive terms of the same president (e.g. Clinton 1st + 2nd -> one
# "Clinton" span) but keeps non-consecutive terms separate (e.g. Trump 1st
# and Trump 2nd are two rows because Biden sits between them).
#
# Returns: label, party, start, end, mid (all x-axis positions).
build_president_labels <- function(year_min, year_max) {
  left  <- year_min - 0.5
  right <- year_max + 0.5

  # Strip " (1st)" / " (2nd)" / etc. to get base president name
  base_name <- gsub(" \\(.+\\)$", "", admins$admin)
  admin_start_x <- as.numeric(format(admins$start, "%Y")) - 0.5
  admin_end_x   <- as.numeric(format(admins$end,   "%Y")) - 0.5

  # Walk admins in chronological order, merging consecutive same-president
  # same-party rows
  labels <- data.frame(label = character(0), party = character(0),
                       start = numeric(0),   end   = numeric(0),
                       stringsAsFactors = FALSE)
  for (i in seq_len(nrow(admins))) {
    n <- nrow(labels)
    if (n > 0 && labels$label[n] == base_name[i] &&
        labels$party[n] == admins$party[i]) {
      labels$end[n] <- admin_end_x[i]   # extend previous span
    } else {
      labels <- rbind(labels, data.frame(
        label = base_name[i], party = admins$party[i],
        start = admin_start_x[i], end = admin_end_x[i],
        stringsAsFactors = FALSE
      ))
    }
  }

  # Clip to plot window and compute label midpoints
  labels$start <- pmax(labels$start, left)
  labels$end   <- pmin(labels$end,   right)
  labels       <- filter(labels, end > left, start < right)
  labels$mid   <- (labels$start + labels$end) / 2

  return(labels)
}
```

R Code For **Figure 1**

```r
# ------------------------------- SCRIPT INFO ---------------------------------
# plot_declarations_per_year.R - Bar chart of major disaster declaration
# requests per presidential year.
#
# Sources fema_helpers.R for data loading + plot helpers.
# -----------------------------------------------------------------------------

rm(list = ls())

source("~/Documents/Classes/PPPA 6021 (Data Visualization)/R Code/fema_helpers.R")

# ---------------------- §1: CONFIG -------------------------------------------

# Plot window (integer presidential-years; Jan-20 split)
year_min <- 1989
year_max <- 2026

# FEMA data files
decl_path <- "~/Documents/Classes/PPPA 6021 (Data Visualization)/Course
Data/FemaWebDisasterDeclarations.csv"
den_path  <- "~/Documents/Classes/PPPA 6021 (Data Visualization)/Course
Data/DeclarationDenials.csv"

# Output file
out_path  <- "~/Downloads/disaster_declarations_per_year.png"

# ---------------------- §2: LOAD + PREPARE -----------------------------------

# Load and clean (keep both approved and denied)
disasters <- load_fema(decl_path, den_path)
disasters <- clean_disasters(disasters)

# Bucket by presidential year and restrict to plot window
disasters$decisionYear <- compute_presidential_year(disasters$decisionDate)
disasters <- filter(disasters, !is.na(decisionYear),
                    decisionYear >= year_min, decisionYear <= year_max)

# Annual counts by status. Order factor so Approved on bottom, Denied on top
disasters$status <- factor(disasters$currentRequestStatus,
                           levels = c("Approved", "Denied"))
annual_counts <- summarise(
  group_by(disasters, decisionYear, status),
  n = n(),
  .groups = "drop"
)

# ---------------------- §3: PLOT SCAFFOLDING ---------------------------------

# Admin shading + president labels (from helpers)
admin_shading    <- build_admin_shading(year_min, year_max)
president_labels <- build_president_labels(year_min, year_max)

# Plot y-axis bounds
y_max   <- 125
label_y <- 130

# Stack colors
status_colors <- c(
  "Approved" = "grey30",
  "Denied"   = "grey60"
)

# Subtitle
subtitle_html <- paste0(
```

```
  "Major disaster declarations per year (<b><span style='color:",
  status_colors["Approved"], ";'>granted</span></b> and <b><span style='color:",
  status_colors["Denied"], ";'>denied</span></b>), ", year_min, "–", year_max, ".",
  "<br><span style='color:gray20; font-size:12pt;'> Background shading ",
  "indicates <span style='color:", PALETTE$Republican, ";'>Republican</span> ",
  "or <span style='color:", PALETTE$Democratic, ";'>Democratic</span> ",
  "presidential administrations.</span>"
)

# ---------------------- §4: BUILD THE GRAPH --------------------------------

graph <- ggplot() +
  # Background shading by administration
  annotate("rect",
           xmin = admin_shading$year_start,
           xmax = admin_shading$year_end,
           ymin = -Inf, ymax = y_max,
           fill = ifelse(admin_shading$party == "R",
                         PALETTE$Republican, PALETTE$Democratic),
           alpha = 0.15) +
  # White dividers between prez terms
  annotate("segment",
           x = admin_shading$year_start, xend = admin_shading$year_start,
           y = 0, yend = y_max,
           color = "white", linewidth = 0.2) +
  # Bars stacked by status (Approved on top, Denied on bottom)
  geom_col(data = annual_counts,
           aes(x = decisionYear, y = n, fill = status),
           width = 0.7) +
  scale_fill_manual(values = status_colors, guide = "none") +
  # President-name labels above the bracket
  annotate("text",
           x = president_labels$mid, y = label_y,
           label = president_labels$label,
           size = 3.5, family = "Helvetica Neue",
           color = ifelse(president_labels$party == "R",
                          PALETTE$Republican, PALETTE$Democratic)) +
  # Bracket showing term spans
  annotate("segment",
           x = president_labels$start, xend = president_labels$end,
           y = y_max, yend = y_max,
           color = ifelse(president_labels$party == "R",
                          PALETTE$Republican, PALETTE$Democratic),
           linewidth = 0.8) +
  # Axes
  scale_x_continuous(breaks = seq(year_min - 1, year_max + 1, by = 4),
                     expand = expansion(mult = c(0.00, 0.02)),
                     limits = c(year_min - 1.5, year_max + 1)) +
  scale_y_continuous(limits = c(0, label_y + 3),
                     breaks = seq(0, y_max, by = 25),
                     expand = expansion(mult = c(0.0, 0.00))) +
  labs(
    title    = "Major Disaster Declaration Requests per Year Have Been Relatively Static",
    subtitle = subtitle_html,
    x        = NULL,
    y        = "Declaration Requests",
    caption  = paste0(
      "Excludes COVID emergency declarations, U.S. territories, tribes, and Freely
Associated States. Annual data split on Inauguration Day (Jan. 20).\n",
      "Data: FEMA National Emergency Management Information System."
    )
  ) +
  coord_cartesian(clip = "off") +
  theme_minimal(base_size = 12, base_family = "Helvetica Neue") +
  theme(
```

```
    legend.position       = "none",
    axis.title.y          = element_blank(),
    axis.text.y           = element_text(size = 10),
    axis.ticks.x          = element_line(color = "grey50", linewidth = 0.4),
    axis.ticks.length.x   = unit(4, "pt"),
    panel.grid.major.x    = element_blank(),
    panel.grid.minor.x    = element_blank(),
    panel.grid.major.y    = element_line(color = "grey90", linewidth = 0.3),
    panel.grid.minor.y    = element_blank(),
    plot.title            = element_text(face = "bold", size = 15, hjust = 0),
    plot.title.position   = "plot",
    plot.caption.position = "plot",
    plot.subtitle         = element_textbox_simple(
      color = "grey20", hjust = 0, margin = margin(b = 14), size = 12
    ),
    plot.caption          = element_text(
      size = 8.0, hjust = 0, margin = margin(t = 14),
      face = "italic", color = "grey30"
    ),
    plot.margin           = margin(t = 20, r = 25, b = 15, l = 20)
  )

graph

ggsave(out_path, graph, width = 10, height = 7.5, dpi = 600)
```

R Code For **Figure 2**

```r
# ------------------------------- SCRIPT INFO --------------------------------
# George Doehne
# PPPA 6021: Data Visualization
# Fall 2026
#
# Description: This program creates a state-level dot density map of
#              presidential major disaster declarations in the United States.
#              Each dot represents 5 declarations, color-coded by disaster type.
#
# ----------------------------- §1: SETUP & DATA -------------------------------

rm(list = ls())

# Load the packages we need:
library(dplyr)
library(ggplot2)
library(tidyverse)
library(tigris)
library(sf)
library(extrafont)

# Read the FEMA disaster declarations CSV file into a data frame
disasters <- read.csv("~/Documents/Classes/PPPA 6021 (Data Visualization)/Course
Data/FemaWebDisasterDeclarations.csv")

# The date column comes in as a long timestamp string (e.g., "2020-01-20T00:00:00.000Z").
# We trim it down to just the first 10 characters ("2020-01-20") and convert it to a Date.
disasters$incidentBeginDate <- substr(disasters$incidentBeginDate, 1, 10)
disasters$incidentBeginDate <- as.Date(disasters$incidentBeginDate)

# Keep Major Disasters, drop Emergencies, Fire Management declarations.
disasters <- filter(disasters, declarationType == "Major Disaster")

# Remove Freely Associated States
disasters <- filter(disasters, !(stateCode %in% c("FM", "MH", "PW")))

# Remove U.S. territories
disasters <- filter(disasters, !(stateCode %in% c("PR", "VI", "GU", "AS", "MP")))

# Remove blanket COVID-19 declarations
disasters <- filter(disasters, !(incidentType == "Biological" & incidentBeginDate ==
as.Date("2020-01-20")))

# Keep only declarations from 1980 onward
disasters <- filter(disasters, incidentBeginDate >= as.Date("1989-01-20"))

# ---------------------------- §2: CATEGORIZE & AGGREGATE --------------------

# ~25 specific incident types down to 7
disasters$disaster_category <- case_when(
  disasters$incidentType %in% c("Severe Storm", "Tornado",
                                "Straight-Line Winds", "Coastal Storm") ~ "Severe Storm",
  disasters$incidentType %in% c("Snowstorm", "Severe Ice Storm",
                                "Winter Storm", "Freezing")             ~ "Winter
Weather",
  disasters$incidentType %in% c("Hurricane", "Tropical Storm",
                                "Tropical Depression", "Typhoon")       ~ "Tropical
System",
  disasters$incidentType %in% c("Flood", "Dam/Levee Break")             ~ "Flood",
  disasters$incidentType %in% c("Fire")                                 ~ "Fire",
  disasters$incidentType %in% c("Earthquake", "Volcanic Eruption",
                                "Tsunami", "Mud/Landslide")             ~ "Geophysical",
  TRUE                                                                  ~ "Other"
)
```

```
# Count how many declarations each state has in each category.
# First group the data by state and category, then count rows in each group.
grouped <- group_by(disasters, stateCode, disaster_category)
state_cat_counts <- summarise(grouped, n = n(), .groups = "drop")

# ---------------------------- §3: GEOMETRIES --------------------------------

# Download simplified state boundary polygons from U.S. Census Bureau.
# cb = TRUE gives cartographic (simplified) boundaries (plot faster)
states_sf <- states(cb = TRUE, year = 2021)

# Move Alaska and Hawaii to inset positions
states_sf <- shift_geometry(states_sf)

# Drop territory FIPS codes from shapefile so only 50 states + DC remain.
# (60 = American Samoa, 66 = Guam, 69 = N. Mariana Islands, 72 = Puerto Rico, 78 = U.S.
Virgin Islands)
states_sf <- states_sf[!states_sf$STATEFP %in% c("60", "66", "69", "72", "78"),]

# Save the cleaned state polygons as base map layer
map_data <- states_sf

# ---------------------------- §4: GENERATE DOTS -----------------------------

# Each dot on the map represents this many declarations.
# With dot_value = 1, every single declaration gets its own dot.
dot_value <- 1

# How far apart (in meters) each dot center is from its neighbors.
# 25000 = 25 km between dot centers.
dot_spacing <- 28000

# Define the order that categories will appear in the legend,
# and assign a color to each one from MetBrewer
cat_levels <- c("Severe Storm", "Flood", "Tropical System",
                "Winter Weather", "Fire", "Geophysical", "Other")

cat_colors <- c("Severe Storm"             = "#f7aa58",   # amber
                "Flood"                    = "#376795",   # deep blue (water)
                "Tropical System"          = "#b695bc",   # olive green (coastal,
tropical)
                "Winter Weather"           = "#aadce0",   # cool grey (cold, muted)
                "Fire"                     = "#e76254",   # red (flame)
                "Geophysical"              = "#6f9969",   # warm orange
                "Other"                    = "grey60")    # neutral gray

# --- OFFSET STATES ---
# Small states (especially in the Northeast) whose dot clusters are pulled out
# to nearby empty space with a line drawn back to the state.
# Offsets are in Albers Equal Area meters — positive x = east, positive y = north.
offset_states <- data.frame(
  STUSPS = c("DC",  "DE",  "RI",  "CT",  "VT",  "NH",  "MD",  "MA"),
  dx     = c( 3.4e5, 5.0e5, 2.8e5, 2.4e5,-3.2e5,-2.0e5, 4.7e5, 3.5e5),
  dy     = c(-4.0e5,-1.8e5,-1.0e5,-2.1e5, 2.5e5, 5.0e5,-2.9e5, 0.9e5),
  stringsAsFactors = FALSE
)

# --- NUDGE STATES ---
# States whose clusters are shifted slightly for readability
nudge_states <- data.frame(
  STUSPS = c("CA",  "ID",  "MI",  "KY",  "TN",  "FL",  "VA",  "NY", "HI", "LA"),
  dx     = c(-0.4e5, 0.0e5, 0.8e5, 0.7e5,  0.0e5, 0.9e5, 0.5e5,  0.3e5,-0.5e5,-0.2e5),
  dy     = c(-0.5e5,-0.7e5,-1.5e5, 0.35e5,-0.2e5, 0.0e5, 0.25e5, 0.2e5,-0.5e5,-0.2e5),
  stringsAsFactors = FALSE
```

```
)

# Save line endpoint data/state codes so lines/labels can be drawn later on map
arrow_data <- data.frame(STUSPS = character(),
                         x_from = numeric(), y_from = numeric(),
                         x_to   = numeric(), y_to   = numeric(),
                         stringsAsFactors = FALSE)

# Loop through every state and generate a cluster of dots at its center.
# Each dot is assigned a disaster category so it gets the right color.
# For offset states, the cluster is shifted and a line is drawn back.
# For nudge states, the cluster is shifted but no line is drawn.
dot_list <- lapply(unique(map_data$STUSPS), function(st) {

  # Look up how many declarations this state has in each category
  st_cats <- filter(state_cat_counts, stateCode == st)

  # If a state has no declarations at all, skip it
  if (nrow(st_cats) == 0) return(NULL)

  # Calculate how many dots this state needs in total
  total <- sum(st_cats$n)
  n_dots_total <- round(total / dot_value)
  if (n_dots_total == 0) return(NULL)

  # Divide the total dots among the categories proportionally.
  # The "largest remainder method" handles rounding so the dot counts
  # add up to the correct total: give each category the floor of its
  # share, then hand out any leftover dots to the categories with the
  # biggest fractional remainders.
  st_cats$raw_share  <- (st_cats$n / total) * n_dots_total
  st_cats$base_dots  <- floor(st_cats$raw_share)
  st_cats$remainder  <- st_cats$raw_share - st_cats$base_dots
  leftover <- n_dots_total - sum(st_cats$base_dots)
  if (leftover > 0) {
    top_idx <- order(st_cats$remainder, decreasing = TRUE)[1:leftover]
    st_cats$base_dots[top_idx] <- st_cats$base_dots[top_idx] + 1
  }
  st_cats$dot_count <- st_cats$base_dots

  # Create a label for every single dot — repeat each category name
  # as many times as that category has dots
  cat_labels <- rep(st_cats$disaster_category, st_cats$dot_count)

  # Sort the labels so the most common category's dots end up in the
  # center of the cluster, with less common categories on the outside
  cat_labels <- cat_labels[order(match(cat_labels,
                                       st_cats$disaster_category[order(st_cats$dot_count,
decreasing = TRUE)]))]

  # Find the geographic center (centroid) of this state's polygon
  state_geom <- map_data[map_data$STUSPS == st, ]
  centroid   <- st_centroid(st_geometry(state_geom))
  cx <- st_coordinates(centroid)[1]  # x coordinate (longitude in projected units)
  cy <- st_coordinates(centroid)[2]  # y coordinate (latitude in projected units)

  # Work out the grid dimensions so we can compute corner positions
  n <- length(cat_labels)
  grid_side <- ceiling(sqrt(n))
  half_extent <- ((grid_side - 1) / 2) * dot_spacing

  # If this is one of the small offset states, shift the dot cluster
  # to a nearby open area. Record the line endpoints — the line will
  # start at either a corner or the midpoint of a side of the cluster,
  # whichever produces a cleaner angle back to the real state.
```

```
if (st %in% offset_states$STUSPS) {
  off <- offset_states[offset_states$STUSPS == st, ]
  new_cx <- cx + off$dx
  new_cy <- cy + off$dy

  # Decide whether the offset line should attach to a corner or the
  # midpoint of a side. We look at the angle from the cluster center
  # back to the real state centroid. If the angle is within ±45° of a
  # cardinal direction (0°/90°/180°/270°), the line is more horizontal
  # or vertical, so we use the nearest side midpoint. Otherwise we
  # use the nearest corner.
  angle <- atan2(cy - new_cy, cx - new_cx)  # angle from cluster to state

  # Normalize angle to [0, 2*pi)
  angle <- angle %% (2 * pi)

  # Check if the angle is within ±45° (pi/4) of a cardinal direction.
  # Cardinal directions in radians: 0, pi/2, pi, 3*pi/2.
  # "Near cardinal" means the line is more horizontal or vertical.
  # The threshold (pi/6 = 30°) biases toward corners; only lines
  # within ±30° of a cardinal direction use a side midpoint.
  near_cardinal <- any(sapply(c(0, pi/2, pi, 3*pi/2, 2*pi), function(a) {
    abs(((angle - a + pi) %% (2 * pi)) - pi) < pi/6
  }))

  if (near_cardinal) {
    # Use the midpoint of the side nearest to the real centroid.
    # Four side midpoints: left, right, top, bottom.
    mid_x <- new_cx + c(-1, 1, 0, 0) * half_extent
    mid_y <- new_cy + c(0, 0, 1, -1) * half_extent
    dists <- sqrt((mid_x - cx)^2 + (mid_y - cy)^2)
    nearest <- which.min(dists)
    from_x <- mid_x[nearest]
    from_y <- mid_y[nearest]
  } else {
    # Use the corner nearest to the real centroid.
    corners_x <- new_cx + c(-1, 1, 1, -1) * half_extent
    corners_y <- new_cy + c(-1, -1, 1, 1) * half_extent
    dists <- sqrt((corners_x - cx)^2 + (corners_y - cy)^2)
    nearest <- which.min(dists)
    from_x <- corners_x[nearest]
    from_y <- corners_y[nearest]
  }

  arrow_data <<- rbind(arrow_data, data.frame(
    STUSPS = st,
    x_from = from_x,
    y_from = from_y,
    x_to   = cx,                    # line ends at the real state centroid
    y_to   = cy,
    stringsAsFactors = FALSE
  ))
  cx <- new_cx
  cy <- new_cy
}

# If this is one of the nudge states, shift the cluster but don't
# draw a line. These are larger states where you just want to
# reposition the cluster slightly for better readability.
if (st %in% nudge_states$STUSPS) {
  nudge <- nudge_states[nudge_states$STUSPS == st, ]
  cx <- cx + nudge$dx
  cy <- cy + nudge$dy
}
```

```
  # Build a square grid of positions centered on the (possibly shifted) centroid.
  # The grid is just big enough to hold all the dots.
  cols <- seq(-(grid_side - 1) / 2, (grid_side - 1) / 2, by = 1)
  rows <- seq(-(grid_side - 1) / 2, (grid_side - 1) / 2, by = 1)
  grid_pts <- expand.grid(col = cols, row = rows)

  # Convert the grid column/row indices into actual map coordinates
  # by multiplying by dot_spacing and adding the centroid position
  grid_pts$x <- cx + grid_pts$col * dot_spacing
  grid_pts$y <- cy + grid_pts$row * dot_spacing

  # The square grid may have more positions than we need dots.
  # Sort by row (top to bottom) then column (left to right) and keep
  # only the first n positions. This fills the grid in a compact
  # square/rectangular block.
  grid_pts <- grid_pts[order(-grid_pts$row, grid_pts$col), ]
  grid_pts <- grid_pts[1:n, ]

  # Tag each grid position with its disaster category label
  grid_pts$disaster_category <- cat_labels

  # Convert the x/y coordinates into a spatial (sf) point object
  # so ggplot can draw them on the map
  pts_sf <- st_as_sf(grid_pts, coords = c("x", "y"), crs = st_crs(map_data))
  return(pts_sf)
})

# Stack all the individual state dot layers into one combined spatial data frame
dots_sf <- do.call(rbind, dot_list[!sapply(dot_list, is.null)])

# Convert the category column to an ordered factor so the legend
# displays categories in our chosen order
dots_sf$disaster_category <- factor(dots_sf$disaster_category, levels = cat_levels)

# For each offset state, figure out where to place the state code label.
# We position it just above the topmost dot in the cluster, with a small
# vertical nudge so the text doesn't overlap the dots.
label_nudge <- 65000  # vertical gap (in meters) between the top dot and the label

label_data <- do.call(rbind, lapply(seq_len(nrow(arrow_data)), function(i) {
  st <- arrow_data$STUSPS[i]
  off <- offset_states[offset_states$STUSPS == st, ]

  # Recompute the offset centroid from the real centroid + offset
  state_geom <- map_data[map_data$STUSPS == st, ]
  centroid   <- st_centroid(st_geometry(state_geom))
  off_cx <- st_coordinates(centroid)[1] + off$dx
  off_cy <- st_coordinates(centroid)[2] + off$dy

  # Figure out how tall the cluster is: the grid extends
  # (grid_side - 1) / 2 * dot_spacing above the center
  st_cats <- filter(state_cat_counts, stateCode == st)
  total <- sum(st_cats$n)
  n_dots <- round(total / dot_value)
  grid_side <- ceiling(sqrt(n_dots))
  cluster_top <- off_cy + ((grid_side - 1) / 2) * dot_spacing

  data.frame(label = st,
             x = off_cx,
             y = cluster_top + label_nudge,
             stringsAsFactors = FALSE)
}))

# ---------------------------- §5: FINAL MAP ----------------------------------
```

```
final_map <- ggplot() +

  # Draw all state polygons filled in light gray with white borders
  geom_sf(data = map_data, fill = "grey90", color = "#FFFFFF", linewidth = 0.5) +

  # Draw thin lines from each offset dot cluster back to its real state.
  # Each line starts at the corner of the cluster nearest to the state,
  # so it doesn't pass behind the dots.
  geom_segment(data = arrow_data,
               aes(x = x_from, y = y_from, xend = x_to, yend = y_to),
               color = "grey60", linewidth = 0.3) +

  # Place a small dot at the state end of each offset line so the
  # reader can see exactly which state the cluster belongs to.
  # Adjust `size` here to make the dot larger or smaller.
  geom_point(data = arrow_data, aes(x = x_to, y = y_to),
             color = "grey60", size = 0.5) +

  # Draw the dots on top of the states, coloring each dot by its disaster category
  geom_sf(data = dots_sf, aes(color = disaster_category), size = 1.25, alpha = 1, shape =
15) +

  # Place the state code (e.g., "VT", "RI") as a label just above each
  # offset cluster so the reader knows which state it belongs to
  geom_text(data = label_data, aes(x = x, y = y, label = label),
            size = 2.5, fontface = "bold", color = "grey30",
            family = "Helvetica Neue") +

  # (Commented out) An optional second state border layer drawn on top of
  # the dots, which can help visually separate overlapping clusters
  # geom_sf(data = map_data, fill = NA, color = "grey60", linewidth = 0.3) +

  # Set the map projection to Albers Equal Area, which is the standard
  # projection for thematic maps of the continental U.S.
  # datum = NA removes the graticule (lat/lon grid lines).
  coord_sf(crs = "+proj=aea +lat_1=29.5 +lat_2=45.5 +lat_0=37.5 +lon_0=-96",
           datum = NA,
           xlim = c(-2.0e6, 2.0e6)) +

  # Reduce whitespace around the edges of the map
  scale_x_continuous(expand = c(0.10, 0)) +
  # The second value adds extra space at the top of the map, giving
  # room between the tallest offset labels (NH, VT) and the legend.
  scale_y_continuous(expand = expansion(mult = c(0.00, 0.05))) +

  # Tell ggplot which colors to use for each disaster category in the dots.
  # drop = FALSE ensures all 7 categories appear in the legend even if a
  # category has zero dots on the map.
  scale_color_manual(values = cat_colors,
                     name   = "Disaster Type",
                     drop   = FALSE) +

  # Add a title and subtitle to the plot
  labs(title    = "Major Disaster Declarations Vary Geographically\nMore by Disaster Types
than by Total Number",
       subtitle = "Federal major disaster declarations by state, 1989–2026. Each square is
one declaration.") +

  # Apply a clean, minimal theme using the PT Sans font
  theme_minimal(base_family = "Helvetica Neue", base_size = 12) +
  theme(
    # Place the color legend horizontally across the top of the map
    legend.position  = "top",
    legend.title     = element_blank(),
    legend.text      = element_text(size = 10, color = "grey20"),
```

```
    legend.key.size  = unit(10, "pt"),
    legend.margin    = margin(b = 5, t = 5, l = 5, r = 5),
    legend.justification = "left",
    legend.background = element_rect(fill = "gray95", color = "#ffffff"),

    # Title styling
    plot.title = element_text(
      face = "bold", hjust = 0.0,
      margin = margin(b = 5),
      size = 15
    ),
    plot.title.position = "plot",
    plot.caption.position = "plot",

    # Subtitle styling
    plot.subtitle = element_text(
      color = "grey20",
      hjust = 0.0,
      margin = margin(b = 10),
      size = 12
    ),

    plot.caption = element_text(size = 8.0, hjust = 0, margin = margin(t = 0),
                                face = "italic", color = "grey50"),

    # Hide axis titles (not needed on a map)
    axis.title.x = element_blank(),
    axis.title.y = element_blank(),

    # Add some breathing room around the edges of the whole plot
    plot.margin = margin(t = 5, r = 0, b = 5, l = 20)
  ) +

  # Force the legend into a single horizontal row and make the legend
  # dots larger (size 3) so they're easier to read than the map dots
  guides(color = guide_legend(nrow = 1, override.aes = list(size = 5))) +

  # Data source citation in lower-right corner
  annotate("text",
           x = 2.3e6, y = -1.60e6,
           label = "Data: FEMA National Emergency Management Information System.",
           size = 3, color = "grey30", fontface = "italic",
           family = "Helvetica Neue", lineheight = 0.9, hjust = 1)

final_map

ggsave("~/Downloads/plot_disaster_distribution.png", final_map,
       width = 10, height = 7.5, dpi = 600)
```

R Code For **Figure 3**

```r
# ------------------------------- SCRIPT INFO ---------------------------------
# plot_count_vs_lean.R - Scatter plot of state-level RAW disaster
# declaration count against 2024 Trump margin. One dot per state, sized
# by 2024 population, colored by lean, with linear regression line.
#
# Companion to plot_rate_vs_lean.R, which shows the per-capita normalized
# version. Raw counts surface the absolute-load story (Texas + Florida +
# California account for huge shares of FEMA's workload regardless of
# partisanship); per-capita rates surface the disaster-likelihood story.
#
# Sources fema_helpers.R for data loading + state populations + lean.
# -----------------------------------------------------------------------------

rm(list = ls())

source("~/Documents/Classes/PPPA 6021 (Data Visualization)/R Code/fema_helpers.R")

library(ggrepel)

# ----------------------- §1: CONFIG ------------------------------------------

# Window for the disaster count (inclusive years; Jan-20 split)
year_min <- 1989
year_max <- 2024

# FEMA data
decl_path <- "~/Documents/Classes/PPPA 6021 (Data Visualization)/Course
Data/FemaWebDisasterDeclarations.csv"
den_path  <- "~/Documents/Classes/PPPA 6021 (Data Visualization)/Course
Data/DeclarationDenials.csv"

# Output file
out_path  <- "~/Downloads/disaster_count_vs_lean.png"

# ----------------------- §2: LOAD + PREPARE ----------------------------------

# Load and clean (keep both approved and denied)
disasters <- load_fema(decl_path, den_path)
disasters <- clean_disasters(disasters)

# disasters <- filter(disasters, currentRequestStatus == "Approved")

# Restrict to window
disasters$decisionYear <- compute_presidential_year(disasters$decisionDate)
disasters <- filter(disasters, !is.na(decisionYear),
                    decisionYear >= year_min, decisionYear <= year_max)

# Per-state declaration request counts (raw, NOT per capita).
# Includes both approved and denied requests.
state_counts <- summarise(
  group_by(disasters, stateCode),
  n_requests = n(),
  .groups = "drop"
)

# Bring in every state from state_population so that any state with zero
# requests still appears as a dot at y = 0
all_states <- data.frame(stateCode = unique(state_population$state),
                         stringsAsFactors = FALSE)
plot_data <- merge(all_states, state_counts, by = "stateCode", all.x = TRUE)
plot_data$n_requests[is.na(plot_data$n_requests)] <- 0

# y-axis is raw requests per year over the window. Dividing by window
# length keeps the y-axis interpretable across different window choices.
```

```
window_years <- year_max - year_min + 1
plot_data$rate <- plot_data$n_requests / window_years

# Continuous Republican lean from state_partisan_lean — using 2024 Trump
# margin (Republican two-party share minus Democratic two-party share).
# Positive values = Trump won the state; negative = Harris won.
if (is.null(state_partisan_lean)) {
  stop("state_partisan_lean not loaded. Run build_state_partisan_lean.R to ",
       "generate state_partisan_lean.csv.")
}
plot_data <- merge(plot_data,
                   state_partisan_lean[, c("state", "pct_gop_2024")],
                   by.x = "stateCode", by.y = "state", all.x = TRUE)
plot_data$rep_share <- plot_data$pct_gop_2024 * 2 - 1   # margin: -1 to +1

# 2024 population for the dot size
size_year <- max(state_population$year)
size_pop  <- filter(state_population, year == size_year)
plot_data <- merge(plot_data, size_pop[, c("state", "population")],
                   by.x = "stateCode", by.y = "state", all.x = TRUE)

# ---------------------- §3: PLOT SCAFFOLDING --------------------------------

# Subtitle
subtitle_html <- paste0(
  "Average number of major disaster declaration requests per year, ",
  year_min, "–", year_max, ". ",
  "Each dot is a state, sized by ", size_year,
  " population and colored by 2024 two-party presidential election margin. "
)

# X-axis breaks every 10 percentage points from -50 to +50
x_breaks <- seq(-0.5, 0.5, by = 0.1)
x_labels <- c("D+50", "D+40", "D+30", "D+20", "D+10",
              "Even",
              "R+10", "R+20", "R+30", "R+40", "R+50")

# ---------------------- §4: BUILD THE GRAPH ---------------------------------

graph <- ggplot(plot_data, aes(x = rep_share, y = rate)) +
  # Corner annotations describing each quadrant. Placed in the data
  # coordinate space using x relative to the [-0.55, 0.55] x-limits and
  # y as a fraction of the data's max rate. Each is two lines via "\n".
  annotate("text",
           x = -0.49, y = 2.49,
           label = "Democratic states with\nmore frequent requests",
           hjust = 0, vjust = 1,
           size = 3.5, lineheight = 1.0, color = "grey50",
           family = "Helvetica Neue") +
  annotate("text",
           x = 0.49, y = 2.49,
           label = "Republican states with\n more frequent requests",
           hjust = 1, vjust = 1,
           size = 3.5, lineheight = 1.0, color = "grey50",
           family = "Helvetica Neue") +
  annotate("text",
           x = -0.49, y = 0.02,
           label = "Democratic states with\nless frequent requests",
           hjust = 0, vjust = 0,
           size = 3.5, lineheight = 1.0, color = "grey50",
           family = "Helvetica Neue") +
  annotate("text",
           x = 0.49, y = 0.02,
           label = "Republican states with\nless frequent requests",
           hjust = 1, vjust = 0,
```

```
          size = 3.5, lineheight = 1.0, color = "grey50",
          family = "Helvetica Neue") +
  # Dots, sized by population, colored by lean
  geom_point(aes(size = population, fill = rep_share),
             shape = 21, color = "white", stroke = 0.6, alpha = 0.85) +
  # State-code labels (non-overlapping via ggrepel)
  geom_text_repel(aes(label = stateCode),
                  size = 3.2, fontface = "bold", color = "grey10",
                  segment.color = "grey60", segment.size = 0.3,
                  box.padding = 0.3, point.padding = 0.4,
                  min.segment.length = 0.3,
                  max.overlaps = 60, seed = 42) +
  # Color scale: solid Democratic blue at D+10 or stronger, solid Republican
  # red at R+10 or stronger, muted purple (#7D3C98) at exactly Even. Values
  # outside [-0.10, +0.10] clamp to the endpoint colors via squish().
  scale_fill_gradientn(
    colors  = c(PALETTE$Democratic, "#7D3C98", PALETTE$Republican),
    values  = c(0, 0.5, 1),
    limits  = c(-0.10, 0.10),
    oob     = scales::squish,
    guide   = "none"
  ) +
  scale_size_continuous(range = c(3, 18), guide = "none") +
  # Axes
  scale_x_continuous(breaks = x_breaks, labels = x_labels,
                     limits = c(-0.55, 0.55),
                     expand = expansion(mult = c(0, 0))) +
  scale_y_continuous(limits = c(0, NA),
                     expand = expansion(mult = c(0.02, 0.08))) +
  labs(
    title    = "States' Partisan Leans Are Somewhat Predictive of\nPast Major Disaster
Declaration Request Frequency",
    subtitle = subtitle_html,
    x        = "2024 presidential election two-party margin (Rep. share minus Dem.
share)",
    y        = "Average requests per year",
    caption  = paste0(
      "Excludes COVID emergency declarations, U.S. territories, tribes, and Freely
Associated States. Annual data split on Inauguration Day (Jan. 20).\n",
      "Data: FEMA National Emergency Management Information System, U.S. Census Bureau,
Fox News Election Results."
    )
  ) +
  theme_minimal(base_size = 12, base_family = "Helvetica Neue") +
  theme(
    legend.position       = "none",
    axis.title.x          = element_text(margin = margin(t = 10), color = "grey30"),
    axis.title.y          = element_text(margin = margin(r = 10), color = "grey30"),
    axis.text             = element_text(color = "grey30"),
    panel.grid.major      = element_line(color = "grey90", linewidth = 0.3),
    panel.grid.minor      = element_blank(),
    plot.title            = element_text(face = "bold", size = 15, hjust = 0),
    plot.title.position   = "plot",
    plot.caption.position = "plot",
    plot.subtitle         = element_textbox_simple(
      color = "grey20", hjust = 0, margin = margin(b = 14), size = 12
    ),
    plot.caption          = element_text(
      size = 8.0, hjust = 0, margin = margin(t = 14),
      face = "italic", color = "grey30"
    ),
    plot.margin           = margin(t = 20, r = 25, b = 15, l = 20)
  )

graph
```

```
ggsave(out_path, graph, width = 7.5, height = 7.5, dpi = 600)
```

R Code For **Figure 4**

```r
# ------------------------------- SCRIPT INFO --------------------------------
# plot_grant_rates_sparklines.R - Three-panel sparkline-style chart showing
# annual grant rate (approvals / total) under all three partisan-state
# methodologies side-by-side: pres_vote, governor, and trifecta.
# ----------------------------------------------------------------------------

rm(list = ls())

source("~/Documents/Classes/PPPA 6021 (Data Visualization)/R Code/fema_helpers.R")

# ----------------------- §1: CONFIG -----------------------------------------

# Plot window (integer presidential-years; Jan-20 split)
year_min <- 1989
year_max <- 2026

# FEMA data files
decl_path <- "~/Documents/Classes/PPPA 6021 (Data Visualization)/Course
Data/FemaWebDisasterDeclarations.csv"
den_path  <- "~/Documents/Classes/PPPA 6021 (Data Visualization)/Course
Data/DeclarationDenials.csv"

# Output file
out_path  <- "~/Downloads/disaster_grant_rates_sparklines2.png"

# Methods, in display order. Each will become one panel.
methods <- c("pres_vote", "governor", "trifecta")

# Panel titles
method_titles <- c(
  pres_vote = "State Partisanship Assigned by Presidential Vote",
  governor  = "State Partisanship Assigned by Governor's Party",
  trifecta  = "State Partisanship Assigned by Governor + Senate Majority"
)

# ----------------------- §2: LOAD + PREPARE ---------------------------------

# Load + clean once. Classification happens per-method below.
disasters_base <- load_fema(decl_path, den_path)
disasters_base <- clean_disasters(disasters_base)

# Bucket by presidential year and restrict to plot window
disasters_base$decisionYear <- compute_presidential_year(disasters_base$decisionDate)
disasters_base <- filter(disasters_base, !is.na(decisionYear),
                         decisionYear >= year_min, decisionYear <= year_max)

# Classify under each method, aggregate (grant rate = approved / total),
# tag with method, and stack
panel_data <- list()
for (m in methods) {
  d <- classify_partisanship(disasters_base, m)
  agg <- summarise(
    group_by(d, decisionYear, state_party),
    n_approved = sum(currentRequestStatus == "Approved"),
    n_total    = n(),
    .groups    = "drop"
  )
  agg$grant_rate <- agg$n_approved / agg$n_total * 100
  agg$method     <- m
  panel_data[[m]] <- agg
}
all_data <- do.call(rbind, panel_data)

# Lock the panel order via factor
```

```
all_data$method <- factor(all_data$method, levels = methods,
                          labels = method_titles[methods])

# Subset for the highlighted endpoints (2025 + 2026, Democratic only)
endpoints <- filter(all_data,
                    decisionYear %in% c(2025, 2026),
                    state_party == "Democratic")

# ---------------------- §3: PLOT SCAFFOLDING --------------------------------

# Admin shading + president labels (rendered per panel via inherit.aes = FALSE)
admin_shading    <- build_admin_shading(year_min, year_max)
president_labels <- build_president_labels(year_min, year_max)

# All methods share the same palette
pal <- get_palette("pres_vote")

# Y-axis bounds: 0–100% with extra room above for the president labels.
y_max <- 100

# President-bracket and label position, expressed as a FRACTION of the
# y-axis range above y_max. Keeping these the same across the wait-times
# and grant-rates scripts ensures consistent visual proportions even
# though their y-axis ranges differ.
bracket_offset_pct <- 0.02   # bracket sits 2% above the data area
label_offset_pct   <- 0.09   # label sits 9% above the data area

bracket_y <- y_max * (1 + bracket_offset_pct)
label_y   <- y_max * (1 + label_offset_pct)

# Subtitle
subtitle_html <- paste0(
  "Overall major disaster declaration grant rate, by year, for
  <b><span style='color:",
  pal["Republican"], ";'>", "Republican states",
  "</span></b> vs. <br><b><span style='color:",
  pal["Democratic"], ";'>", "Democratic states",
  "</span></b>", ", ", year_min, "–", year_max, ".",
  "<span style='color:gray20; font-size:12pt;'> Background shading indicates <span
style='color:",
  pal["Republican"], ";'>Republican</span> or <span style='color:",
  pal["Democratic"], ";'>Democratic</span> presidential administrations. ", "</span>"
)

# ---------------------- §4: BUILD THE GRAPH ---------------------------------

graph <- ggplot(all_data, aes(x = decisionYear, y = grant_rate)) +
  # Background shading (fills each panel up to y_max)
  geom_rect(data = admin_shading, inherit.aes = FALSE,
            aes(xmin = year_start, xmax = year_end,
                ymin = -Inf, ymax = y_max,
                fill = party),
            alpha = 0.15) +
  scale_fill_manual(values = c("R" = PALETTE$Republican,
                               "D" = PALETTE$Democratic),
                    guide = "none") +
  # White dividers between same-party back-to-back terms
  geom_segment(data = admin_shading, inherit.aes = FALSE,
               aes(x = year_start, xend = year_start,
                   y = 0, yend = y_max),
               color = "white", linewidth = 0.2) +
  # President-name labels above each panel
  geom_text(data = president_labels, inherit.aes = FALSE,
            aes(x = mid, y = label_y, label = label,
                color = party),
```

```
            size = 3.5, family = "Helvetica Neue",
            show.legend = FALSE) +
  # Bracket showing each presidency's term span
  geom_segment(data = president_labels, inherit.aes = FALSE,
               aes(x = start, xend = end,
                   y = bracket_y, yend = bracket_y,
                   color = party),
               linewidth = 0.8, show.legend = FALSE) +
  # Partisan lines
  geom_line(aes(color = state_party),
            linewidth = 1.5, alpha = 0.85) +
  # Highlight Democratic endpoints (2025 + 2026): bigger dots + percent labels
  geom_point(data = endpoints,
             aes(color = state_party),
             size = 3.0) +
  geom_text(data = endpoints,
            aes(label = sprintf("%.0f%%", grant_rate),
                color = state_party,
                x = decisionYear + ifelse(decisionYear == 2025, -0.5, 0.5),
                y = grant_rate + ifelse(decisionYear == 2025, -0.0, -5.0)),
            vjust = 1.0,
            hjust = 1.0,
            size = 4.0, fontface = "bold", family = "Helvetica Neue",
            show.legend = FALSE) +
  # Combined color scale: state_party uses pal, party (R/D from admin shading
  # tables) maps to the same hex codes.
  scale_color_manual(values = c(pal,
                                "R" = PALETTE$Republican,
                                "D" = PALETTE$Democratic),
                     guide = "none") +
  # Three vertically-stacked sparkline panels, shared y-axis
  facet_wrap(~ method, ncol = 1, strip.position = "top") +
  # Axes
  scale_x_continuous(breaks = seq(year_min - 1, year_max + 1, by = 4),
                     expand = expansion(mult = c(0.00, 0.02)),
                     limits = c(year_min - 1.5, year_max + 1.5)) +
  scale_y_continuous(labels = function(x) paste0(x, "%"),
                     limits = c(0, y_max * 1.13),
                     breaks = c(0, 25, 50, 75, 100),
                     expand = expansion(mult = c(0.0, 0.0))) +
  labs(
    title    = "By Any Metric, Major Disaster Declaration\nGrant Rates Are Diverging by
Party",
    subtitle = subtitle_html,
    x        = NULL,
    y        = NULL,
    caption  = paste0(
      "Excludes COVID emergency declarations, U.S. territories, tribes, and Freely
Associated States. Annual data split on Inauguration Day (Jan. 20).\n",
      "Data: FEMA National Emergency Management Information System, MIT Election Data,
Ballotpedia, @unitedstates/congress-legislators GitHub."
    )
  ) +
  coord_cartesian(clip = "off") +
  theme_minimal(base_size = 12, base_family = "Helvetica Neue") +
  theme(
    legend.position       = "none",
    axis.title            = element_blank(),
    axis.text.x           = element_text(color = "grey30"),
    axis.text.y           = element_text(color = "grey50", size = 10),
    axis.ticks.x          = element_line(color = "grey50", linewidth = 0.4),
    axis.ticks.length.x   = unit(4, "pt"),
    panel.grid.major.x    = element_blank(),
    panel.grid.minor      = element_blank(),
    panel.grid.major.y    = element_line(color = "grey90", linewidth = 0.3),
```

```
    panel.spacing.y       = unit(28, "pt"),
    strip.text            = element_text(face = "bold", size = 12, hjust = 0),
    strip.background      = element_blank(),
    plot.title            = element_text(face = "bold", size = 15, hjust = 0),
    plot.title.position   = "plot",
    plot.caption.position = "plot",
    plot.subtitle         = element_textbox_simple(
      color = "grey20", hjust = 0, margin = margin(b = 14 ), size = 12
    ),
    plot.caption          = element_text(
      size = 8.0, hjust = 0, margin = margin(t = 14),
      face = "italic", color = "grey30"
    ),
    plot.margin           = margin(t = 20, r = 25, b = 15, l = 20)
  )

graph

ggsave(out_path, graph, width = 7.5, height = 10, dpi = 600)
```

R Code For **Figure 5**

```
# ------------------------------- SCRIPT INFO ---------------------------------
# plot_grant_rates_opposite_party.R - Bar graph comparing the grant rate
# (approved / total) for "opposite-party" pairings
#
# Four bars:
#   1. Republican states under Democratic presidents,  1989-01-20 to 2025-01-20
#   2. Democratic states under Republican presidents,  1989-01-20 to 2025-01-20
#   3. Republican states under Trump II,               2025-01-20 to present
#   4. Democratic states under Trump II,               2025-01-20 to present
#
# -----------------------------------------------------------------------------

rm(list = ls())

source("~/Documents/Classes/PPPA 6021 (Data Visualization)/R Code/fema_helpers.R")

# ---------------------- §1: CONFIG ---------------------------------------

# Partisanship methodology: "pres_vote" | "governor" | "trifecta"
method <- "pres_vote"

# FEMA data files
decl_path <- "~/Documents/Classes/PPPA 6021 (Data Visualization)/Course
Data/FemaWebDisasterDeclarations.csv"
den_path  <- "~/Documents/Classes/PPPA 6021 (Data Visualization)/Course
Data/DeclarationDenials.csv"

# Output file
out_path  <- paste0("~/Downloads/disaster_grant_rates_opposite_party_", method, ".png")

# Period boundaries (Jan 20 of each year matches inauguration day)
hist_start <- as.Date("1989-01-20")    # Bush Sr. start
hist_end   <- as.Date("2025-01-20")    # End of Biden / start of Trump II
trump2_end <- as.Date("2026-05-01")    # "Present" — adjust as data updates

# ---------------------- §2: LOAD + PREPARE -------------------------------

# Load, clean, classify
disasters <- load_fema(decl_path, den_path)
disasters <- clean_disasters(disasters)
disasters <- classify_partisanship(disasters, method)

# Drop rows where decision date is missing (can't bucket them by period)
disasters <- filter(disasters, !is.na(decisionDate))

# Tag each row with the sitting president's party on the decision date.
# admins is loaded from fema_helpers.R; admin party "D"/"R" maps to full name.
disasters$president_party <- NA_character_
for (i in 1:nrow(admins)) {
  mask <- !is.na(disasters$decisionDate) &
          disasters$decisionDate >= admins$start[i] &
          disasters$decisionDate <  admins$end[i]
  disasters$president_party[mask] <- ifelse(
    admins$party[i] == "R", "Republican", "Democratic"
  )
}
disasters <- filter(disasters, !is.na(president_party))

# ---------------------- §3: BUILD THE FOUR PERIOD/PAIRING SUBSETS ----------

# Helper: take subset of `disasters` matching a date window + state_party +
# president_party combination, return grant rate (approvals / total) and n.
period_grant_rate <- function(start, end, state_pty, pres_pty) {
  d <- filter(disasters,
```

```
              decisionDate >= start, decisionDate < end,
              state_party  == state_pty,
              president_party == pres_pty)
  data.frame(
    grant_rate = sum(d$currentRequestStatus == "Approved") / nrow(d) * 100,
    n          = nrow(d)
  )
}

# Build the four bars in chosen display order
bars <- rbind(
  cbind(
    label  = "Republican states\nunder Dem. presidents\n(1989–2025)",
    state  = "Republican",
    period = "Historical",
    period_grant_rate(hist_start, hist_end, "Republican", "Democratic")
  ),
  cbind(
    label  = "Democratic states\nunder Rep. presidents\n(1989–2025)",
    state  = "Democratic",
    period = "Historical",
    period_grant_rate(hist_start, hist_end, "Democratic", "Republican")
  ),
  cbind(
    label  = "Republican states\nunder Trump II\n(2025–pres.)",
    state  = "Republican",
    period = "Trump II",
    period_grant_rate(hist_end, trump2_end, "Republican", "Republican")
  ),
  cbind(
    label  = "Democratic states\nunder Trump II\n(2025–pres.)",
    state  = "Democratic",
    period = "Trump II",
    period_grant_rate(hist_end, trump2_end, "Democratic", "Republican")
  )
)

# Lock the bar order
bars$label <- factor(bars$label, levels = bars$label)

# ---------------------- §4: PLOT SCAFFOLDING --------------------------------

pal    <- get_palette(method)
labels <- get_method_labels(method)

# Y-axis upper bound (always 0–100% for grant rates, with room for labels)
y_max  <- 110

# Subtitle (literal v16-style HTML)
subtitle_html <- paste0(
  "Grant rate (approvals as a share of all requests) for major disaster ",
  "declarations, comparing <b><span style='color:", pal["Republican"], ";'>",
  labels$r_phrase, "</span></b> and <b><span style='color:",
  pal["Democratic"], ";'>", labels$d_phrase, "</span></b> ",
  "when the sitting president was of the opposite party to Trump's second term.",
  "<span style='color:gray20; font-size:12pt;'> ", labels$method_note
)

# ---------------------- §5: BUILD THE GRAPH ---------------------------------

graph <- ggplot(bars, aes(x = label, y = grant_rate, fill = state)) +
  geom_col(width = 0.65) +
  # Numeric labels above each bar
  geom_text(aes(label = sprintf("%.1f%%", grant_rate)),
            vjust = -0.3, size = 5.5, fontface = "bold",
```

```
           family = "Helvetica Neue", color = "grey20") +
  # Vertical separator between historical and Trump II groups
  annotate("segment", x = 2.5, xend = 2.5,
           y = 0, yend = y_max * 0.92,
           color = "grey50", linewidth = 0.4, linetype = "dashed") +
  scale_fill_manual(values = pal, guide = "none") +
  scale_y_continuous(labels = function(x) paste0(x, "%"),
                     limits = c(0, y_max),
                     breaks = seq(0, 100, by = 25),
                     expand = expansion(mult = c(0, 0))) +
  labs(
    title    = "Trump Is Unusually Hostile to Opposite-Party\nMajor Disaster Declaration
Requests",
    subtitle = subtitle_html,
    x        = NULL,
    y        = NULL,
    caption  = paste0(
      "Excludes COVID emergency declarations, U.S. territories, tribes, and Freely
Associated States. Annual data split on Inauguration Day (Jan. 20).\n",
      "Data: FEMA National Emergency Management Information System, MIT Election Data."
    )
  ) +
  coord_cartesian(clip = "off") +
  theme_minimal(base_size = 12, base_family = "Helvetica Neue") +
  theme(
    legend.position       = "none",
    axis.title            = element_blank(),
    axis.text.x           = element_text(size = 10, lineheight = 1.0,
                                         color = "grey20"),
    axis.text.y           = element_text(color = "grey30"),
    axis.ticks            = element_blank(),
    panel.grid.major.x    = element_blank(),
    panel.grid.minor      = element_blank(),
    panel.grid.major.y    = element_line(color = "grey90", linewidth = 0.3),
    plot.title            = element_text(face = "bold", size = 15, hjust = 0),
    plot.title.position   = "plot",
    plot.caption.position = "plot",
    plot.subtitle         = element_textbox_simple(
      color = "grey20", hjust = 0, margin = margin(b = 14), size = 12
    ),
    plot.caption          = element_text(
      size = 8.0, hjust = 0, margin = margin(t = 14),
      face = "italic", color = "grey30"
    ),
    plot.margin           = margin(t = 20, r = 25, b = 15, l = 20)
  )

graph

ggsave(out_path, graph, width = 7.5, height = 5, dpi = 600)
```

R Code For **Figure 6**

```r
# ------------------------------- SCRIPT INFO ---------------------------------
# plot_wait_times_sparklines.R - Three-panel sparkline-style chart showing
# average days from request to decision under all three partisan-state
# methodologies side-by-side: pres_vote, governor, and trifecta.
# -----------------------------------------------------------------------------

rm(list = ls())

source("~/Documents/Classes/PPPA 6021 (Data Visualization)/R Code/fema_helpers.R")

# ----------------------- §1: CONFIG ------------------------------------------

# Plot window (integer presidential-years; Jan-20 split)
year_min <- 1989
year_max <- 2026

# FEMA data files
decl_path <- "~/Documents/Classes/PPPA 6021 (Data Visualization)/Course
Data/FemaWebDisasterDeclarations.csv"
den_path  <- "~/Documents/Classes/PPPA 6021 (Data Visualization)/Course
Data/DeclarationDenials.csv"

# Output file
out_path  <- "~/Downloads/disaster_wait_times_sparklines2.png"

# Methods, in display order. Each becomes one panel.
methods <- c("pres_vote", "governor", "trifecta")

# Panel titles
method_titles <- c(
  pres_vote = "State Partisanship Assigned by Presidential Vote",
  governor  = "State Partisanship Assigned by Governor's Party",
  trifecta  = "State Partisanship Assigned by Governor + Senate Majority"
)

# ----------------------- §2: LOAD + PREPARE ----------------------------------

# Load + clean
disasters_base <- load_fema(decl_path, den_path)
disasters_base <- clean_disasters(disasters_base)

# Wait time + year columns added to base
disasters_base$wait_days <- as.numeric(disasters_base$decisionDate -
                                         disasters_base$declarationRequestDate)
disasters_base <- filter(disasters_base, !is.na(wait_days), wait_days >= 0)
disasters_base$decisionYear <- compute_presidential_year(disasters_base$decisionDate)
disasters_base <- filter(disasters_base, !is.na(decisionYear),
                         decisionYear >= year_min, decisionYear <= year_max)

# Classify under each method, aggregate (mean wait_days), tag with method, stack
panel_data <- list()
for (m in methods) {
  d <- classify_partisanship(disasters_base, m)
  agg <- summarise(
    group_by(d, decisionYear, state_party),
    avg_days = mean(wait_days),
    n        = n(),
    .groups  = "drop"
  )
  agg$method <- m
  panel_data[[m]] <- agg
}
all_data <- do.call(rbind, panel_data)
```

```
# Lock the panel order via factor
all_data$method <- factor(all_data$method, levels = methods,
                          labels = method_titles[methods])

# Subset for the highlighted endpoints (2025 + 2026, Democratic only)
endpoints <- filter(all_data,
                    decisionYear %in% c(2025, 2026),
                    state_party == "Democratic")

# ---------------------- §3: PLOT SCAFFOLDING --------------------------------

# Admin shading + president labels (rendered per panel via inherit.aes = FALSE)
admin_shading    <- build_admin_shading(year_min, year_max)
president_labels <- build_president_labels(year_min, year_max)

# All methods share the same palette
pal <- get_palette("pres_vote")

# Y-axis upper bound (shared across panels for direct comparison).
y_max <- 120

# President-bracket and label position, expressed as a FRACTION of the
# y-axis range above y_max. Keeping these the same across the wait-times
# and grant-rates scripts ensures consistent visual proportions even
# though their y-axis ranges differ.
bracket_offset_pct <- 0.02   # bracket sits 2% above the data area
label_offset_pct   <- 0.09   # label sits 9% above the data area

bracket_y <- y_max * (1 + bracket_offset_pct)
label_y   <- y_max * (1 + label_offset_pct)

# Subtitle (literal v16-style HTML)
subtitle_html <- paste0(
  "Average wait, by year, for major disaster declaration decisions for
  <b><span style='color:",
  pal["Republican"], ";'>", "Republican states",
  "</span></b> vs. <b><span style='color:",
  pal["Democratic"], ";'>", "Democratic states",
  "</span></b>", ", ", year_min, "–", year_max, ".",
  "<span style='color:gray20; font-size:12pt;'> Background shading indicates <span
style='color:",
  pal["Republican"], ";'>Republican</span> or <span style='color:",
  pal["Democratic"], ";'>Democratic</span> presidential administrations. ", "</span>"
)

# ---------------------- §4: BUILD THE GRAPH ---------------------------------

graph <- ggplot(all_data, aes(x = decisionYear, y = avg_days)) +
  # Background shading (fills each panel up to y_max)
  geom_rect(data = admin_shading, inherit.aes = FALSE,
            aes(xmin = year_start, xmax = year_end,
                ymin = -Inf, ymax = y_max,
                fill = party),
            alpha = 0.15) +
  scale_fill_manual(values = c("R" = PALETTE$Republican,
                               "D" = PALETTE$Democratic),
                    guide = "none") +
  # White dividers between same-party back-to-back terms
  geom_segment(data = admin_shading, inherit.aes = FALSE,
               aes(x = year_start, xend = year_start,
                   y = 0, yend = y_max),
               color = "white", linewidth = 0.2) +
  # President-name labels above each panel
  geom_text(data = president_labels, inherit.aes = FALSE,
            aes(x = mid, y = label_y, label = label,
```

```
              color = party),
          size = 3.5, family = "Helvetica Neue",
          show.legend = FALSE) +
  # Bracket showing each presidency's term span
  geom_segment(data = president_labels, inherit.aes = FALSE,
               aes(x = start, xend = end,
                   y = bracket_y, yend = bracket_y,
                   color = party),
               linewidth = 0.8, show.legend = FALSE) +
  # Partisan lines
  geom_line(aes(color = state_party),
            linewidth = 1.5, alpha = 0.85) +
  # Dots for recent Dem data points
  geom_point(data = endpoints,
             aes(color = state_party),
             size = 3.0) +
  # Labels for recent Dem data points
  geom_text(data = endpoints,
            aes(label = sprintf("%.0f days", avg_days),
                color = state_party),
            vjust = 0.1,
            hjust = 1.2,
            size = 4.0, fontface = "bold", family = "Helvetica Neue",
            show.legend = FALSE) +
  # Combined color scale: state_party uses pal, party (R/D from admin shading
  # tables) maps to the same hex codes. Both render correctly.
  scale_color_manual(values = c(pal,
                                "R" = PALETTE$Republican,
                                "D" = PALETTE$Democratic),
                     guide = "none") +
  # Three vertically-stacked sparkline panels, shared y-axis
  facet_wrap(~ method, ncol = 1, strip.position = "top") +
  # Axes — minimal, sparkline-style
  scale_x_continuous(breaks = seq(year_min - 1, year_max + 1, by = 4),
                     expand = expansion(mult = c(0.00, 0.02)),
                     limits = c(year_min - 1.5, year_max + 1.5)) +
  scale_y_continuous(labels = function(x) paste0(x, " d"),
                     limits = c(0, y_max * 1.13),
                     breaks = c(0, 30, 60, 90, 120),
                     expand = expansion(mult = c(0.0, 0.0))) +
  labs(
    title    = "By Any Metric, Major Disaster Declaration\nDecision Wait Times Are
Diverging by Party",
    subtitle = subtitle_html,
    x        = NULL,
    y        = NULL,
    caption  = paste0(
      "Excludes COVID emergency declarations, U.S. territories, tribes, and Freely
Associated States. Annual data split on Inauguration Day (Jan. 20).\n",
      "Data: FEMA National Emergency Management Information System, MIT Election Data,
Ballotpedia, @unitedstates/congress-legislators GitHub."
    )
  ) +
  coord_cartesian(clip = "off") +
  theme_minimal(base_size = 12, base_family = "Helvetica Neue") +
  theme(
    legend.position       = "none",
    axis.title            = element_blank(),
    axis.text.x           = element_text(color = "grey30"),
    axis.text.y           = element_text(color = "grey50", size = 10),
    axis.ticks.x          = element_line(color = "grey50", linewidth = 0.4),
    axis.ticks.length.x   = unit(4, "pt"),
    panel.grid.major.x    = element_blank(),
    panel.grid.minor      = element_blank(),
    panel.grid.major.y    = element_line(color = "grey90", linewidth = 0.3),
```

```
    panel.spacing.y       = unit(28, "pt"),
    strip.text            = element_text(face = "bold", size = 12, hjust = 0),
    strip.background      = element_blank(),
    plot.title            = element_text(face = "bold", size = 15, hjust = 0),
    plot.title.position   = "plot",
    plot.caption.position = "plot",
    plot.subtitle         = element_textbox_simple(
      color = "grey20", hjust = 0, margin = margin(b = 14), size = 12
    ),
    plot.caption          = element_text(
      size = 8.0, hjust = 0, margin = margin(t = 14),
      face = "italic", color = "grey30"
    ),
    plot.margin           = margin(t = 20, r = 25, b = 15, l = 20)
  )

graph

ggsave(out_path, graph, width = 7.5, height = 10, dpi = 600)
```

R Code For **Figure 7**

```
# ------------------------------- SCRIPT INFO ---------------------------------
# plot_wait_times_opposite_party.R - Bar graph comparing average days from
# request to decision for "opposite-party" pairings
#
# Four bars:
#   1. Republican states under Democratic presidents,  1989-01-20 to 2025-01-20
#   2. Democratic states under Republican presidents,  1989-01-20 to 2025-01-20
#   3. Republican states under Trump II,               2025-01-20 to present
#   4. Democratic states under Trump II,               2025-01-20 to present
# -----------------------------------------------------------------------------

rm(list = ls())

source("~/Documents/Classes/PPPA 6021 (Data Visualization)/R Code/fema_helpers.R")

# ----------------------- §1: CONFIG ------------------------------------------

# Partisanship methodology: "pres_vote" | "governor" | "trifecta"
method <- "governor"

# FEMA data files
decl_path <- "~/Documents/Classes/PPPA 6021 (Data Visualization)/Course
Data/FemaWebDisasterDeclarations.csv"
den_path  <- "~/Documents/Classes/PPPA 6021 (Data Visualization)/Course
Data/DeclarationDenials.csv"

# Output file
out_path  <- paste0("~/Downloads/disaster_wait_times_opposite_party_", method, ".png")

# Period boundaries (Jan 20 of each year matches inauguration day)
hist_start <- as.Date("1989-01-20")    # Bush Sr. start
hist_end   <- as.Date("2025-01-20")    # End of Biden / start of Trump II
trump2_end <- as.Date("2026-05-01")    # "Present" — adjust as data updates

# ----------------------- §2: LOAD + PREPARE ----------------------------------

# Load, clean, classify
disasters <- load_fema(decl_path, den_path)
disasters <- clean_disasters(disasters)
disasters <- classify_partisanship(disasters, method)

# Wait time = days from request to decision; drop missing/negative
disasters$wait_days <- as.numeric(disasters$decisionDate -
                                  disasters$declarationRequestDate)
disasters <- filter(disasters, !is.na(wait_days), wait_days >= 0)

# Tag each row with the sitting president's party on the decision date.
# admins is loaded from fema_helpers.R; admin party "D"/"R" maps to full name.
disasters$president_party <- NA_character_
for (i in 1:nrow(admins)) {
  mask <- !is.na(disasters$decisionDate) &
          disasters$decisionDate >= admins$start[i] &
          disasters$decisionDate <  admins$end[i]
  disasters$president_party[mask] <- ifelse(
    admins$party[i] == "R", "Republican", "Democratic"
  )
}
disasters <- filter(disasters, !is.na(president_party))

# ----------------------- §3: BUILD THE FOUR PERIOD/PAIRING SUBSETS ----------

# Helper: take subset of `disasters` matching a date window + state_party +
# president_party combination, return mean wait_days and n.
period_mean <- function(start, end, state_pty, pres_pty) {
```

```
  d <- filter(disasters,
              decisionDate >= start, decisionDate < end,
              state_party  == state_pty,
              president_party == pres_pty)
  data.frame(
    avg_days = mean(d$wait_days),
    n        = nrow(d)
  )
}

# Build the four bars in chosen display order
bars <- rbind(
  cbind(
    label  = "Republican states\nunder Dem. presidents\n(1989–2025)",
    state  = "Republican",
    period = "Historical",
    period_mean(hist_start, hist_end, "Republican", "Democratic")
  ),
  cbind(
    label  = "Democratic states\nunder Rep. presidents\n(1989–2025)",
    state  = "Democratic",
    period = "Historical",
    period_mean(hist_start, hist_end, "Democratic", "Republican")
  ),
  cbind(
    label  = "Republican states\nunder Trump II\n(2025–pres.)",
    state  = "Republican",
    period = "Trump II",
    period_mean(hist_end, trump2_end, "Republican", "Republican")
  ),
  cbind(
    label  = "Democratic states\nunder Trump II\n(2025–pres.)",
    state  = "Democratic",
    period = "Trump II",
    period_mean(hist_end, trump2_end, "Democratic", "Republican")
  )
)

# Lock the bar order
bars$label <- factor(bars$label, levels = bars$label)

# ---------------------- §4: PLOT SCAFFOLDING --------------------------------

pal    <- get_palette(method)
labels <- get_method_labels(method)

# Y-axis upper bound, with room above for value labels
y_max  <- max(bars$avg_days) * 1.18

# Subtitle (literal v16-style HTML)
subtitle_html <- paste0(
  "Average days from request to decision for major disaster declarations, ",
  "comparing <b><span style='color:", pal["Republican"], ";'>",
  labels$r_phrase, "</span></b> and <b><span style='color:",
  pal["Democratic"], ";'>", labels$d_phrase, "</span></b> ",
  "when the sitting president was of the opposite party.",
  "<span style='color:gray20; font-size:12pt;'> ", labels$method_note
)

# ---------------------- §5: BUILD THE GRAPH ---------------------------------

graph <- ggplot(bars, aes(x = label, y = avg_days, fill = state)) +
  geom_col(width = 0.65) +
  # Numeric labels above each bar
  geom_text(aes(label = sprintf("%.1f days", avg_days)),
```

```
            vjust = -0.3, size = 5.5, fontface = "bold",
            family = "Helvetica Neue", color = "grey20") +
  # Sample size labels inside each bar (small, white text)
  geom_text(aes(label = paste0("n = ", n), y = avg_days * 0.05),
            vjust = 0, size = 2.8, family = "PT Sans",
            color = "white", fontface = "italic") +
  # Vertical separator between historical and Trump II groups
  annotate("segment", x = 2.5, xend = 2.5,
           y = 0, yend = y_max * 0.92,
           color = "grey50", linewidth = 0.4, linetype = "dashed") +
  scale_fill_manual(values = pal, guide = "none") +
  scale_y_continuous(labels = function(x) paste0(x, " days"),
                     limits = c(0, y_max),
                     expand = expansion(mult = c(0, 0))) +
  labs(
    title    = "Trump Is Unusually Slow to Answer Major Disaster Declaration Requests",
    subtitle = subtitle_html,
    x        = NULL,
    y        = NULL,
    caption  = paste0(
      "Excludes COVID emergency declarations, U.S. territories, tribes, and Freely
Associated States. Annual data split on Inauguration Day (Jan. 20).\n",
      "Data: FEMA National Emergency Management Information System, MIT Election Data."
    )
  ) +
  coord_cartesian(clip = "off") +
  theme_minimal(base_size = 12, base_family = "Helvetica Neue") +
  theme(
    legend.position       = "none",
    axis.title            = element_blank(),
    axis.text.x           = element_text(size = 10, lineheight = 1.0,
                                         color = "grey20"),
    axis.text.y           = element_text(color = "grey30"),
    axis.ticks            = element_blank(),
    panel.grid.major.x    = element_blank(),
    panel.grid.minor      = element_blank(),
    panel.grid.major.y    = element_line(color = "grey90", linewidth = 0.3),
    plot.title            = element_text(face = "bold", size = 15, hjust = 0),
    plot.title.position   = "plot",
    plot.caption.position = "plot",
    plot.subtitle         = element_textbox_simple(
      color = "grey20", hjust = 0, margin = margin(b = 14), size = 12
    ),
    plot.caption          = element_text(
      size = 8.0, hjust = 0, margin = margin(t = 14),
      face = "italic", color = "grey30"
    ),
    plot.margin           = margin(t = 20, r = 25, b = 15, l = 20)
  )

graph

ggsave(out_path, graph, width = 7.5, height = 5, dpi = 600)
```

R Code For **Figure 8**

```
# ------------------------------- SCRIPT INFO ---------------------------------
# plot_disaster_scale.R - Two-panel chart testing whether disasters approved
# under Trump II are systematically more "catastrophic" than historical baseline
#
#   Top panel: median counties per disaster, by year
#   Bottom:    mean programs (IH/IA/PA/HM) activated per disaster, by year
#
# Trump II (2025+) bars are highlighted; everything else is the baseline.
#
# Sources fema_helpers.R for plot helpers + palette
# -----------------------------------------------------------------------------

rm(list = ls())

source("~/Documents/Classes/PPPA 6021 (Data Visualization)/R Code/fema_helpers.R")

# ---------------------- §1: CONFIG ------------------------------------------

year_min <- 1989
year_max <- 2026

# County-level summaries CSV (NOT FemaWebDisasterDeclarations)
summary_path <- "~/Documents/Classes/PPPA 6021 (Data Visualization)/Course
Data/DisasterDeclarationsSummaries.csv"

out_path <- "~/Downloads/disaster_scale_panels.png"

# ---------------------- §2: LOAD + PREPARE ----------------------------------

raw <- read.csv(summary_path)

# Parse declarationDate (FEMA timestamp format)
raw$declarationDate <- as.Date(substr(raw$declarationDate, 1, 10))

# Filter: Major Disasters only; drop FAS, territories
raw <- filter(raw, declarationType == "DR")
raw <- filter_out(raw, state %in% c("FM", "MH", "PW",
                                    "PR", "VI", "GU", "AS", "MP"))

# Bucket by presidential year (Jan-20 split)
raw$decisionYear <- compute_presidential_year(raw$declarationDate)
raw <- filter(raw, !is.na(decisionYear),
                   decisionYear >= year_min, decisionYear <= year_max)

# Build county FIPS for distinct-county counts
raw$fipsCounty <- paste0(formatC(as.integer(raw$fipsStateCode),
                                 width = 2, flag = "0"),
                         formatC(as.integer(raw$fipsCountyCode),
                                 width = 3, flag = "0"))

# Per-disaster aggregation: how many counties; how many programs activated.
# Programs are 0/1 flags; take any() across the disaster's rows for
# each program (a program is "activated" if ANY county under that disaster
# had it flagged), then sum the four to get a 0–4 score.
per_disaster <- summarise(
  group_by(raw, disasterNumber, decisionYear, incidentType),
  counties = n_distinct(fipsCounty),
  ih       = as.integer(any(ihProgramDeclared == 1)),
  ia       = as.integer(any(iaProgramDeclared == 1)),
  pa       = as.integer(any(paProgramDeclared == 1)),
  hm       = as.integer(any(hmProgramDeclared == 1)),
  .groups  = "drop"
)
per_disaster$programs <- per_disaster$ih + per_disaster$ia +
```

```
                          per_disaster$pa + per_disaster$hm

# Annual summary stats
annual <- summarise(
  group_by(per_disaster, decisionYear),
  median_counties  = median(counties),
  mean_programs    = mean(programs),
  n_disasters      = n(),
  .groups          = "drop"
)

# Trump II flag for highlighting
annual$is_trump2 <- annual$decisionYear >= 2025

# ----------------------- §3: PLOT SCAFFOLDING --------------------------------

admin_shading    <- build_admin_shading(year_min, year_max)
president_labels <- build_president_labels(year_min, year_max)

# Highlight color for Trump II bars ( Republican red)
highlight_color <- PALETTE$Republican
neutral_color   <- "grey40"

# Reshape annual into long form for facet_wrap
long <- rbind(
  data.frame(decisionYear = annual$decisionYear,
             value        = annual$median_counties,
             is_trump2    = annual$is_trump2,
             panel        = "Median counties per disaster"),
  data.frame(decisionYear = annual$decisionYear,
             value        = annual$mean_programs,
             is_trump2    = annual$is_trump2,
             panel        = "Mean programs activated per disaster (out of 4)")
)
long$panel <- factor(long$panel,
                     levels = c("Median counties per disaster",
                                "Mean programs activated per disaster (out of 4)"))

# Per-panel y-bound (proportional bracket placement, matching the sparkline
# convention)
panel_ymax <- c("Median counties per disaster" = max(annual$median_counties) * 1.15,
                "Mean programs activated per disaster (out of 4)" = 4)
bracket_offset_pct <- 0.02
label_offset_pct   <- 0.09

# Build a per-panel "scaffolding" data frame for the admin shading + president
# bracket positions, since they need different y values per panel.
build_scaffold <- function() {
  out <- list()
  for (p in names(panel_ymax)) {
    y_max     <- panel_ymax[[p]]
    bracket_y <- y_max * (1 + bracket_offset_pct)
    label_y   <- y_max * (1 + label_offset_pct)

    sh <- admin_shading
    sh$panel     <- p
    sh$y_max     <- y_max

    pl <- president_labels
    pl$panel     <- p
    pl$bracket_y <- bracket_y
    pl$label_y   <- label_y

    out[[p]] <- list(shading = sh, labels = pl, y_max = y_max,
                     y_top = y_max * 1.13)
```

```
  }
  out
}
scaffold <- build_scaffold()

shading_long  <- do.call(rbind, lapply(scaffold, function(x) x$shading))
labels_long   <- do.call(rbind, lapply(scaffold, function(x) x$labels))
shading_long$panel <- factor(shading_long$panel, levels = levels(long$panel))
labels_long$panel  <- factor(labels_long$panel,  levels = levels(long$panel))

# y-axis caps per panel (data frame with y limits for facet)
panel_lims <- data.frame(
  panel = factor(names(panel_ymax), levels = levels(long$panel)),
  y_top = sapply(scaffold, function(x) x$y_top)
)

# Subtitle
subtitle_html <- paste0(
  "Two FEMA data measures of disaster scale, by year, ",
  year_min, "–", year_max, ". <b><span style='color:",
  highlight_color, ";'>Trump II</span></b> bars are highlighted. ",
  "<span style='color:gray20; font-size:11pt;'> Background shading indicates <span
style='color:",
  PALETTE$Republican, ";'>Republican</span> or <span style='color:",
  PALETTE$Democratic, ";'>Democratic</span> presidential administrations. ",
  "Programs are FEMA's four assistance categories: Individual ",
  "Households (IH), Individual Assistance (IA), Public Assistance (PA), and ",
  "Hazard Mitigation (HM).</span>"
)

# ----------------------- §4: BUILD THE GRAPH ---------------------------------

graph <- ggplot(long, aes(x = decisionYear, y = value)) +
  # Background shading (per panel)
  geom_rect(data = shading_long, inherit.aes = FALSE,
            aes(xmin = year_start, xmax = year_end,
                ymin = -Inf, ymax = y_max,
                fill = party),
            alpha = 0.15) +
  scale_fill_manual(values = c("R" = PALETTE$Republican,
                               "D" = PALETTE$Democratic),
                    guide = "none") +
  # White dividers between same-party back-to-back terms
  geom_segment(data = shading_long, inherit.aes = FALSE,
               aes(x = year_start, xend = year_start,
                   y = 0, yend = y_max),
               color = "white", linewidth = 0.2) +
  # President-name labels above each panel
  geom_text(data = labels_long, inherit.aes = FALSE,
            aes(x = mid, y = label_y, label = label,
                color = party),
            size = 3.5, family = "Helvetica Neue",
            show.legend = FALSE) +
  # Bracket showing each presidency's term span
  geom_segment(data = labels_long, inherit.aes = FALSE,
               aes(x = start, xend = end,
                   y = bracket_y, yend = bracket_y,
                   color = party),
               linewidth = 0.8, show.legend = FALSE) +
  # Bars (highlighted Trump II in red, others in neutral grey)
  geom_col(aes(fill = is_trump2),
           width = 0.7, alpha = 0.95) +
  # Override fill for the bar layer
  scale_fill_manual(values = c("FALSE" = neutral_color,
                               "TRUE"  = highlight_color),
```

```
                    aesthetics = "fill",
                    guide = "none",
                    name = NULL) +
  scale_color_manual(values = c("R" = PALETTE$Republican,
                                "D" = PALETTE$Democratic),
                     guide = "none") +
  facet_wrap(~ panel, ncol = 1, scales = "free_y", strip.position = "top") +
  # X-axis
  scale_x_continuous(breaks = seq(year_min - 1, year_max + 1, by = 4),
                     expand = expansion(mult = c(0.00, 0.02)),
                     limits = c(year_min - 1.5, year_max + 1.5)) +
  scale_y_continuous(expand = expansion(mult = c(0.0, 0.0))) +
  labs(
    title    = "Major Disaster Declarations Under Trump\nHaven't Been for 'Bigger'
Disasters",
    subtitle = subtitle_html,
    x        = NULL,
    y        = NULL,
    caption  = paste0(
      "Excludes U.S. territories, tribes, and ",
      "Freely Associated States. Annual data split on Inauguration Day (Jan. 20).\n",
      "Data: FEMA National Emergency Management Information System."
    )
  ) +
  coord_cartesian(clip = "off") +
  theme_minimal(base_size = 12, base_family = "Helvetica Neue") +
  theme(
    legend.position       = "none",
    axis.title            = element_blank(),
    axis.text.x           = element_text(color = "grey30"),
    axis.text.y           = element_text(color = "grey50", size = 10),
    axis.ticks.x          = element_line(color = "grey50", linewidth = 0.4),
    axis.ticks.length.x   = unit(4, "pt"),
    panel.grid.major.x    = element_blank(),
    panel.grid.minor      = element_blank(),
    panel.grid.major.y    = element_line(color = "grey90", linewidth = 0.3),
    panel.spacing.y       = unit(28, "pt"),
    strip.text            = element_text(face = "bold", size = 12,
                                         hjust = 0, color = "grey20",
                                         margin = margin(b = 8)),
    strip.background      = element_blank(),
    plot.title            = element_text(face = "bold", size = 15, hjust = 0,
                                         margin = margin(b = 8)),
    plot.title.position   = "plot",
    plot.caption.position = "plot",
    plot.subtitle         = element_textbox_simple(
      color = "grey20", hjust = 0, margin = margin(b = 14), size = 11
    ),
    plot.caption          = element_text(
      size = 8.0, hjust = 0, margin = margin(t = 14),
      face = "italic", color = "grey30"
    ),
    plot.margin           = margin(t = 20, r = 25, b = 15, l = 20)
  )

graph

ggsave(out_path, graph, width = 7.5, height = 7.5, dpi = 600)
```