

# NEH Grants from the 2000s Data Analysis

Brian A. Danielak

## Table of contents

<b>1</b>	<b>Executive Summary</b>	<b>2</b>
<b>2</b>	<b>Loading and Cleaning the Data</b>	<b>2</b>
2.1	Loading Necessary Libraries . . . . .	2
2.2	Loading Grant Data from Downloaded XML . . . . .	3
2.3	Inspecting the Data . . . . .	4
<b>3</b>	<b>Research Question: What's the Per-Capita Total Amount Awarded for Each State?</b>	<b>5</b>
3.1	Analysis Overview . . . . .	5
3.2	Loading Census Data with <code>tidycensus</code> . . . . .	5
3.3	Determining Which NEH Award States/Territories Aren't In the Census Data .	6
3.4	Augmenting the <code>state</code> Dataset with Census State/Territory Abbreviations . .	7
3.5	How Much NEH Awards Data Are We Losing Because of Missing US States/Regions in the Census? . . . . .	7
3.6	Augmenting 2000s Grants Data with 2010 Census Data . . . . .	8
3.7	Computing the Total Amount Awarded . . . . .	8
3.8	Computing the Dollars Awarded Per-Capita in Each State . . . . .	9
3.9	Visualizing Awards Per-Capita in Each State . . . . .	10
<b>4</b>	<b>Analysis and Interpretation</b>	<b>15</b>
<b>5</b>	<b>Limitations</b>	<b>16</b>
5.1	Dollar Amounts are Unadjusted for Inflation . . . . .	16
5.2	We Excluded Certain US Regions and Territories . . . . .	16
5.3	We used the total award plus the total matching, as opposed to just the amount the NEH spent on the award. . . . .	17
5.4	There are Limits to the Reproducibility of This Document . . . . .	17
<b>6</b>	<b>Technology Used</b>	<b>17</b>
<b>7</b>	<b>Sources</b>	<b>17</b>

# 1 Executive Summary

```
# The values used in the executive Summary are actually computed later in the
# document, but in order to demonstrate how I would
# typically write a Quarto computable document, I've lifted those
# values here.
#
# In a typical workflow, the analysis would prepare the data objects to be used
# by the Qmd document.

national_per_capita_award <- 3.52
dc_per_capita_award <- 42.4786
```

We examine briefly how the total NEH award dollars and matching funds are distributed on a per-state, per-capita basis. Breaking the data down per-capita helps correct for the distortion effects of very low-population and very high-population states because states are considered proportionately, rather than by their population. There are three major findings:

1. The national average is \$3.52 US dollars total awarded per person.
2. Washington, DC received \$42.48 dollars per person, the highest in the nation at 3 times the next highest state.
3. When we censor DC, we note that several states with low populations (e.g., Alaska and Wyoming) nevertheless have high awards dollars per-capita.
4. New York's percentage share of total dollars awarded leads at around 12%, with California at roughly 6%. This seems disproportionate given their per capita dollars awarded.

## 2 Loading and Cleaning the Data

### 2.1 Loading Necessary Libraries

```
suppressPackageStartupMessages(
{
  library(tidycensus) # to grab population data from the Census
  library(forcats) # to reorder data
  library(here) # to handle relative paths to files
  library(dplyr) # for data manipulation
  library(ggplot2) # for data visualization
  library(knitr) # To use kable for pretty tables
  library(scales) # to format numbers
```

```

library(vroom) # to quickly load data
library(xml2) # to parse the downloaded XML data
library(usmap) # to plot quantities on a geographic US map
}
)

# Load US Census API Key (http://api.census.gov/data/key_signup.html)
# Committing the API key would be a security hazard, so we
# read it from an external, non-version-controlled file.
# We use `read_lines()` so as not to get a trailing newline
# character when reading the file.
my_census_api_key <- readr::read_lines(
  here("us-census-api-key.txt"),
  n_max = 1
)

```

## 2.2 Loading Grant Data from Downloaded XML

```

load_grants_data <- function(
  year,
  path_to_xml = sprintf(
    here("data-raw/NEH_Grants%s_Flat/NEH_Grants%ss_Flat.xml"),
    year,
    year
  )
) {

  # First, we check whether we've cached the cleaned data.
  if (file.exists(here("data-cleaned/grants-data-cleaned.tsv"))) {

    # If it exists, we vroom it and implicitly return it
    vroom(here("data-cleaned/grants-data-cleaned.tsv"))

  } else {
    grants_raw <- read_xml(path_to_xml) |>
      as_list()

    grants_tibble <- grants_raw$Grants |>

```

```

    lapply(unlist) |>
    bind_rows()

    # Write the rectangularized data out so we can load *it* next time.
    # `vroom::write` returns its data argument, so that's what the function
    # will return if it takes the `else` branch
    grants_tibble |>
      transmute_all(unlist) |>
      vroom_write(here("data-cleaned/grants-data-cleaned.tsv"))
  }
}

# I downloaded and unzipped the 2000s grant data
# https://securegrants.neh.gov/open/data/NEH_Grants2000s_Flat.zip
grants_2000s <- load_grants_data(2000)

```

## 2.3 Inspecting the Data

First, let's get an idea of what the data looks like.

```

grants_2000s |>
  dplyr::glimpse()

```

```

Rows: 9,402
Columns: 32
$ ApplicantType      <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
$ Institution        <chr> "Mississippi Action for Community Edu., Inc.", "~
$ OrganizationType   <chr> "Community-Level Organization", "Four-Year Colle~
$ InstCity           <chr> "Greenville", "Milledgeville", "Boise", "New Yor~
$ InstState          <chr> "MS", "GA", "ID", "NY", "PA", "NE", "RI", "NY", ~
$ InstPostalCode     <chr> "38701", "31061-3375", "83702-6053", "10002-3102~
$ InstCountry        <chr> "USA", "USA", "USA", "USA", "USA", "USA", "USA", ~
$ CongressionalDistrict <dbl> 2, 10, 2, 7, 18, 1, 1, 25, 5, 4, 25, 20, 20, 7, ~
$ Latitude           <chr> "33.38353", "33.080870", "43.615340", "40.718640~
$ Longitude          <chr> "-91.05397", "-83.229410", "-116.194150", "-73.9~
$ CouncilDate        <date> 2000-11-01, 2001-11-01, 2001-11-01, 2001-11-01, ~
$ YearAwarded        <dbl> 2000, 2001, 2001, 2001, 2001, 2001, 2002, 2001, ~
$ ProjectTitle       <chr> "The Delta Cultural History Project", "Getting t~
$ Program            <chr> "Challenge Grants", "Challenge Grants", "Challen~
$ Division           <chr> "Challenge Programs", "Challenge Programs", "Cha~

```

```

$ ApprovedOutright      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
$ ApprovedMatching      <dbl> 180000, 500000, 150000, 500000, 300000, 450000, ~
$ AwardOutright         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
$ AwardMatching         <dbl> 140000.0, 300000.0, 150000.0, 500000.0, 300000.0~
$ OriginalAmount        <dbl> 140000.0, 300000.0, 150000.0, 500000.0, 300000.0~
$ SupplementAmount      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
$ BeginGrant            <date> 2002-06-01, 1999-12-01, 2000-12-01, 2000-12-01,~
$ EndGrant              <date> 2007-12-31, 2006-07-31, 2006-12-31, 2006-07-31,~
$ ProjectDesc           <chr> "No description", "No description", "No descript~
$ ToSupport             <chr> "Endowment to fund a position of Director of Cul~
$ PrimaryDiscipline     <chr> "Music History and Criticism", "U.S. Regional St~
$ SupplementCount       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
$ ParticipantCount      <dbl> 2, 2, 1, 1, 2, 1, 4, 1, 1, 1, 1, 1, 1, 1, 3, 1, ~
$ Participants          <chr> "Ruby Buck [Project Director]; Catherine Courtne~
$ DisciplineCount       <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
$ Disciplines           <chr> "Music History and Criticism", "U.S. Regional St~
$ Supplements           <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~

```

### 3 Research Question: What's the Per-Capita Total Amount Awarded for Each State?

#### 3.1 Analysis Overview

To answer our research question, we're going to:

1. Get census data via an API call
2. Join census populations with existing awards data
3. Visualize the Awards Per-Capita in each state

#### 3.2 Loading Census Data with tidycensus

```

# Tell tidycensus to use my API key
census_api_key(my_census_api_key, install = FALSE)

load_census_data <-
  function(census_data = here("data-cleaned/2010-census.tsv")) {
    # We don't want to keep access the API
    # every time our code runs,
    # so if we already have the data, vroom it and return it.
  }

```

```

    if (file.exists(census_data)) {
      vroom(census_data)
    } else {
      # Otherwise,
      # we access the API for it, vroom it, and return it
      # I had to choose a census year to match the 2000s NEH grant data,
      # so I settled on 2010.
      get_decennial(
        geography = "state",
        variables = "P001001",
        year = 2010
      ) |>
      vroom_write(census_data)
    }
  }
}

state_populations <- load_census_data()

states <- state_populations |>
  rename(
    census_state = NAME,
    population = value
  )

```

### 3.3 Determining Which NEH Award States/Territories Aren't In the Census Data

Unfortunately, R's built-in `state` dataset only has data for the 50 states and not other US territories/regions. So, let's find out which census regions/territories don't exist in R's built-in set. Then we can add them.

```

# First we grab and store all the unique values in the Census states.
census_state_names <- states |>
  distinct(census_state) |>
  pull()

# Then, we use them to index into state abbreviations to find out which
# aren't in R's built-in dataset using setdiff() saving for later possible use
states_in_census_but_not_in_built_in_dataset <-
  setdiff(census_state_names, state.name)

```

```
states_in_census_but_not_in_built_in_dataset
```

```
[1] "District of Columbia" "Puerto Rico"
```

### 3.4 Augmenting the state Dataset with Census State/Territory Abbreviations

OK, now that we know those states, we can augment the existing state dataset with those abbreviations.

```
census_states <- state.name

# Use a named vector to more easily index state names by their abbreviations
names(census_states) <- state.abb

# Now we can add DC and Puerto Rico
census_states <- c(
  census_states,
  DC = "District of Columbia",
  PR = "Puerto Rico"
)
```

### 3.5 How Much NEH Awards Data Are We Losing Because of Missing US States/Regions in the Census?

We use our newly-augmented `census_states` to see how many grants would get dropped because we don't have US Census populations for them.

```
`%notin%` <- Negate(`%in%`)
grants_with_no_population_info <-
  grants_2000s |>
  # We're looking for NEH award regions *not* in our census data
  filter((InstState %notin% names(census_states)))

number_of_grants_with_no_population_data <- nrow(grants_with_no_population_info)

percentage_of_grants_without_population_data <-
  round(nrow(grants_with_no_population_info) / nrow(grants_2000s) * 100, 2)
```

The original dataset had 9402 rows. There are 95 grants in states with no populations. So, if we drop them, we're losing 1.01 percent of our data.

### 3.6 Augmenting 2000s Grants Data with 2010 Census Data

Now that we know the consequences of dropping data, we can try joining the census data on our NEH awards data.

```
grants_with_population_data <-
  grants_2000s |>
  filter(InstState %in% names(census_states))

# You can't subset within a base-R pipe, so we create this variable
indexed_census_states <- census_states[grants_with_population_data$InstState]

# We index into the census states vector to grab full state names from abbreviations,
# then we augment the dataframe with that new column
grants_with_population_data <-
  grants_with_population_data |>
  mutate(
    census_state = indexed_census_states,
  ) |>
  left_join(states, by = "census_state")
```

### 3.7 Computing the Total Amount Awarded

Here, we augment the data again with my best sense of the **total amount** of a grant awarded. In my thinking, that would be the **AwardOutright** plus matching funds (**AwardMatching**).<sup>1</sup>

```
grants_with_population_data <-
  grants_with_population_data |>
  mutate(grant_total = AwardOutright + AwardMatching)

glimpse(grants_with_population_data)
```

Rows: 9,307

Columns: 37

\$ ApplicantType	<dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
\$ Institution	<chr> "Mississippi Action for Community Edu., Inc.", "~
\$ OrganizationType	<chr> "Community-Level Organization", "Four-Year Colle~
\$ InstCity	<chr> "Greenville", "Milledgeville", "Boise", "New Yor~

---

<sup>1</sup>This computation is based on my understanding of the column names as described in the dataset's data dictionary.



```

$ InstState          <chr> "MS", "GA", "ID", "NY", "PA", "NE", "RI", "NY", ~
$ InstPostalCode     <chr> "38701", "31061-3375", "83702-6053", "10002-3102~
$ InstCountry        <chr> "USA", "USA", "USA", "USA", "USA", "USA", "USA",~
$ CongressionalDistrict <dbl> 2, 10, 2, 7, 18, 1, 1, 25, 5, 4, 25, 20, 20, 7, ~
$ Latitude           <chr> "33.38353", "33.080870", "43.615340", "40.718640~
$ Longitude          <chr> "-91.05397", "-83.229410", "-116.194150", "-73.9~
$ CouncilDate        <date> 2000-11-01, 2001-11-01, 2001-11-01, 2001-11-01,~
$ YearAwarded        <dbl> 2000, 2001, 2001, 2001, 2001, 2001, 2002, 2001, ~
$ ProjectTitle       <chr> "The Delta Cultural History Project", "Getting t~
$ Program            <chr> "Challenge Grants", "Challenge Grants", "Challen~
$ Division           <chr> "Challenge Programs", "Challenge Programs", "Cha~
$ ApprovedOutright   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
$ ApprovedMatching   <dbl> 180000, 500000, 150000, 500000, 300000, 450000, ~
$ AwardOutright      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
$ AwardMatching      <dbl> 140000.0, 300000.0, 150000.0, 500000.0, 300000.0~
$ OriginalAmount     <dbl> 140000.0, 300000.0, 150000.0, 500000.0, 300000.0~
$ SupplementAmount   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
$ BeginGrant         <date> 2002-06-01, 1999-12-01, 2000-12-01, 2000-12-01,~
$ EndGrant           <date> 2007-12-31, 2006-07-31, 2006-12-31, 2006-07-31,~
$ ProjectDesc        <chr> "No description", "No description", "No descript~
$ ToSupport          <chr> "Endowment to fund a position of Director of Cul~
$ PrimaryDiscipline  <chr> "Music History and Criticism", "U.S. Regional St~
$ SupplementCount     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
$ ParticipantCount   <dbl> 2, 2, 1, 1, 2, 1, 4, 1, 1, 1, 1, 1, 1, 1, 3, 1, ~
$ Participants       <chr> "Ruby Buck [Project Director]; Catherine Courtne~
$ DisciplineCount     <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
$ Disciplines        <chr> "Music History and Criticism", "U.S. Regional St~
$ Supplements        <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
$ census_state       <chr> "Mississippi", "Georgia", "Idaho", "New York", "~
$ GEOID              <chr> "28", "13", "16", "36", "42", "31", "44", "36", ~
$ variable           <chr> "P001001", "P001001", "P001001", "P001001", "P00~
$ population         <dbl> 2967297, 9687653, 1567582, 19378102, 12702379, 1~
$ grant_total        <dbl> 140000.0, 300000.0, 150000.0, 500000.0, 300000.0~

```

### 3.8 Computing the Dollars Awarded Per-Capita in Each State

Next we:

1. Group by state
2. Sum the total awards for each state
3. Create a new column dividng those totals by the state population
4. Check the output to make sure the transformation worked

Then we check the first few rows to make sure the transformation worked.

```
awards_per_capita_by_state <-  
  grants_with_population_data |>  
  select(census_state, grant_total, population) |>  
  group_by(census_state, population) |>  
  summarize(  
    total_awarded_to_state = (sum(grant_total))  
  ) |>  
  mutate(per_capita_award = total_awarded_to_state / population) |>  
  arrange(desc(per_capita_award))
```

``summarise()`` has grouped output by `'census_state'`. You can override using the ``.groups`` argument.

```
glimpse(awards_per_capita_by_state)
```

Rows: 52

Columns: 4

Groups: census\_state [52]

```
$ census_state      <chr> "District of Columbia", "Vermont", "Alaska", "M~  
$ population        <dbl> 601723, 625741, 710231, 6547629, 1052567, 13283~  
$ total_awarded_to_state <dbl> 25560353, 9389525, 8987774, 75701190, 11914422,~  
$ per_capita_award   <dbl> 42.478605, 15.005450, 12.654719, 11.561619, 11.~
```

### 3.9 Visualizing Awards Per-Capita in Each State

Now we want to be able to compare states to each other based on the number of dollars the NEH awarded in grants per-capita in each state. Dividing the total awarded amount to the state by the population gives us a normalized number of dollars spent per person. In other words, we ask, “how much was awarded in each state on a per-person basis?”. Computing things this way helps us identify states where NEH awards are high vs. low without being distorted by state population.

```
national_per_capita_award <- (  
  sum(awards_per_capita_by_state$total_awarded_to_state) /  
  sum(awards_per_capita_by_state$population)  
)  
  
national_award_total <- sum(awards_per_capita_by_state$total_awarded_to_state)
```

```

awards_per_capita_by_state |>
  ggplot() +
  geom_point(
    aes(x = forcats::fct_reorder(census_state, per_capita_award),
        y = per_capita_award,
        size = population,
        color = ((total_awarded_to_state / national_award_total) * 100)
    )
  ) +
  geom_hline(
    yintercept = national_per_capita_award,
    color = "darkorange",
    alpha = 0.75,
  ) +

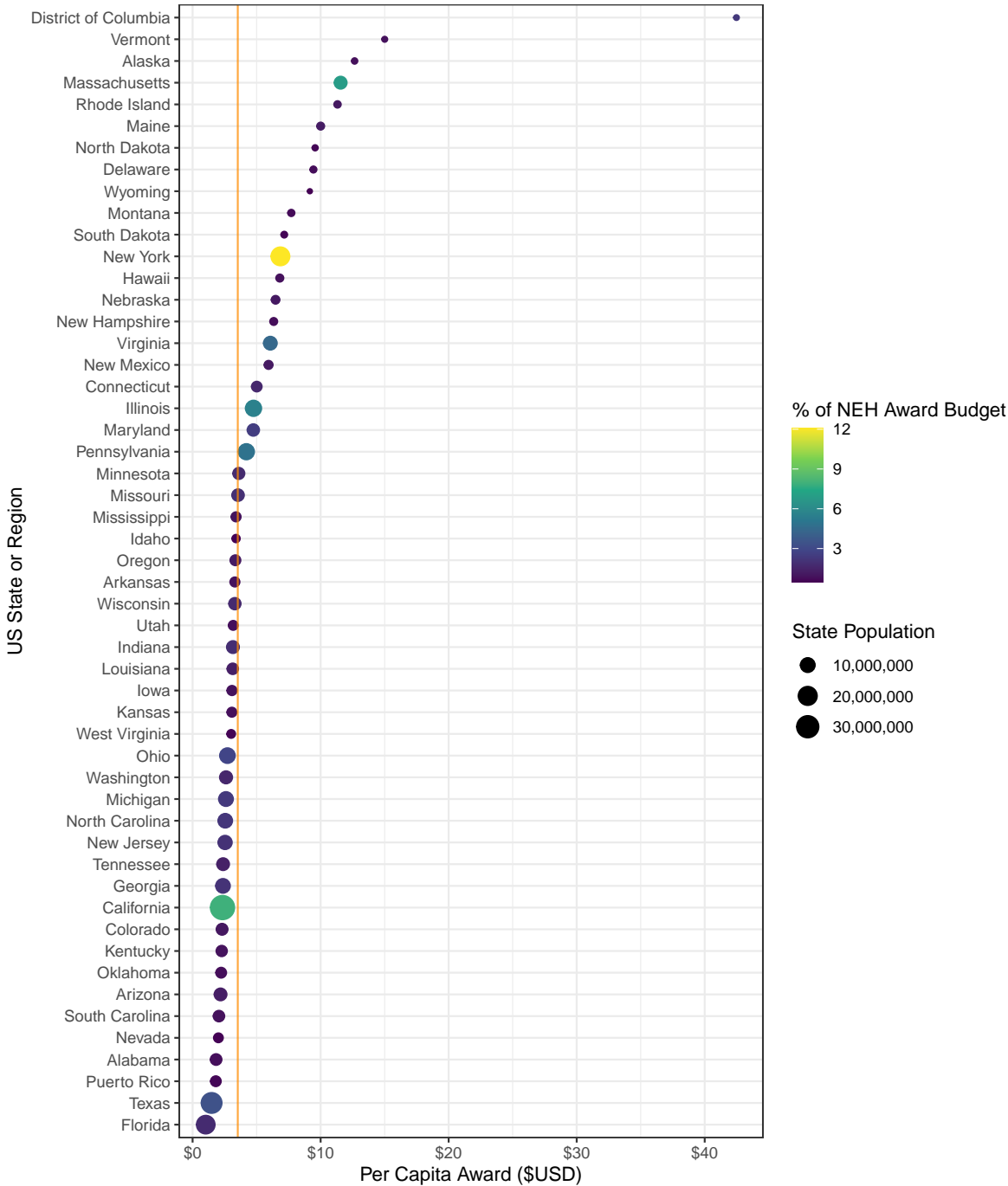
  # We customize the scales and labels so we don't get
  # raw R variable names
  scale_size_continuous(
    name = "State Population",
    labels = scales::label_comma()
  ) +
  scale_color_viridis_c(name = "% of NEH Award Budget") +
  scale_y_continuous(
    name = "Per Capita Award ($USD)",
    labels = scales::label_dollar()
  ) +

  # We make all labels more human-readable
  labs(
    x = "US State or Region",
    title = "Exploring Total Award Dollars Per-Capita",
    subtitle = "Broken down by US States and Territories",
    caption = "The mean national per-capita NEH award amount is indicated by the orange ve
  ) +

  # Finally, we flip the coordinates because it's easier to read
  # left-to right
  coord_flip() +
  theme_bw()

```

# Exploring Total Award Dollars Per-Capita Broken down by US States and Territories

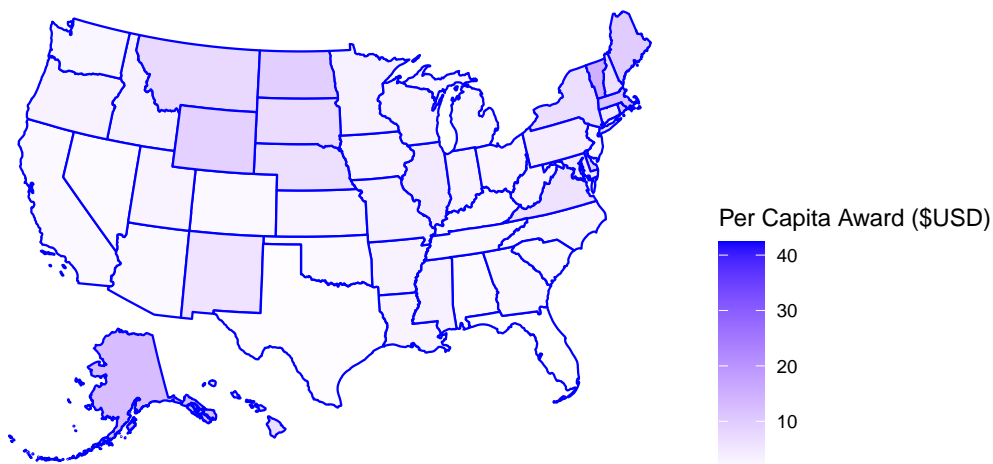


```

per_capita_by_state <- tibble(
  n = pull(awards_per_capita_by_state, per_capita_award),
  state = pull(awards_per_capita_by_state, census_state)
)

plot_usmap(
  data = per_capita_by_state,
  values = "n",
  color = "blue"
) +
  scale_fill_continuous(low = "white", high = "blue", name = "Per Capita Award ($USD)") +
  theme(legend.position = "right")

```



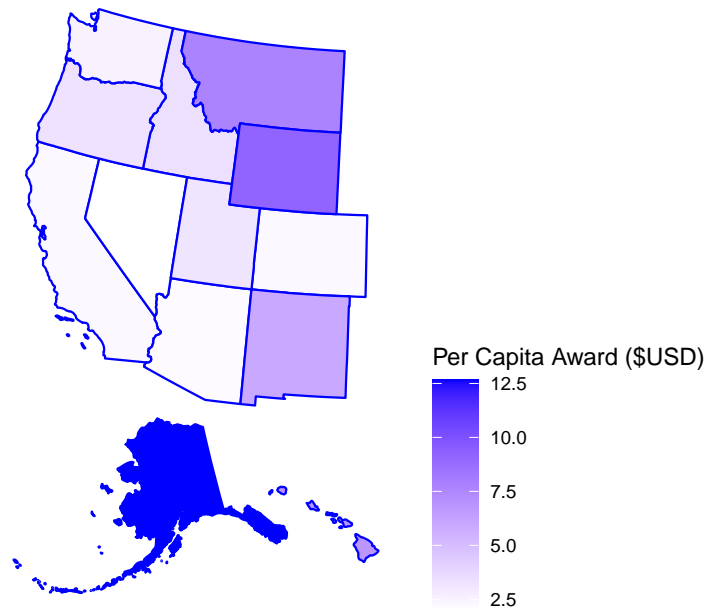
The extreme DC value appears to be masked in a typical USA map plot due to the small size of the District.

```

plot_usmap(
  data = per_capita_by_state,
  values = "n",
  color = "blue",
  include = .west_region

```

```
) +
  scale_fill_continuous(low = "white", high = "blue", name = "Per Capita Award ($USD)") +
  theme(legend.position = "right")
```



Here, we see that when adjusted for NEH dollars received per-capita, larger area states appear not to be over represented.

```
state_abbrvs <- intersect(names(census_states), .west_region)

states <- census_states[state_abbrvs]

per_capita_by_state |>
  filter(state %in% states) |>
  mutate(dollars_per_capita = round(n, 2)) |>
  select(state, dollars_per_capita) |>
  kable()
```

state	dollars_per_capita
Alaska	12.65
Wyoming	9.16

state	dollars_per_capita
Montana	7.71
Hawaii	6.81
New Mexico	5.93
Idaho	3.40
Oregon	3.34
Utah	3.17
Washington	2.61
California	2.34
Colorado	2.30
Arizona	2.18
Nevada	2.02

## 4 Analysis and Interpretation

```
# We extract awards per capita for outliers
dc_per_capita_award <-
  awards_per_capita_by_state |>
  filter(census_state == "District of Columbia") |>
  pull()

florida_per_capita_award <-
  awards_per_capita_by_state |>
  filter(census_state == "Florida") |>
  pull()
```

At \$42.48, DC receives more NEH dollars awarded per-person number than any other state in the US. Could this be because DC has such a small population? That guess seems somewhat unlikely because we tried to correct for population with our per-person counts. Could it be that the number of awards in DC is simply greater than the number of awards in other states? One possible speculation is that DC has more museums and humanities venues per person than any other state/territory in the country. Another is unlike states, which can receive some grant funding through their state governments, DC isn't a state, so perhaps all its grant requests are federal and must through the NEH.

When we explore geographic maps of per-capita funding, we see that almost all regions look low, and it's difficult to find where the high per-capita regions are. This difficulty is caused by DC being so high in per-capita award dollars and also so small in area.

We note most states are tightly grouped around \$3.53 US dollars per person. Low outliers are Florida and Texas at roughly \$1.03 dollars per person, while at the upper end Washington

DC has roughly \$42.48 US dollars per person, about 3 times the amount of grant dollars per person than the next highest state of Vermont.

Finally, when we look at which states which represent the greatest *percentage* overall of NEH + matching award dollars, we note that New York comes in around 12%, nearly twice the amount of the next state with the highest share, which is California at roughly 6%. These figures suggest those states are taking a higher than average share of grant dollars given the amount of dollars spent per-person.

## 5 Limitations

### 5.1 Dollar Amounts are Unadjusted for Inflation

The NEH grants used in our dataset cover a 10-year period, but we didn't augment the data with inflation figures, so our analysis assumes today's value of the dollar.

### 5.2 We Excluded Certain US Regions and Territories

Not all parts of the United States are represented in this analysis. While we only excluded 1.01% of the observations in the data, that still excludes entire US regions and territories such as American Samoa and the U.S. Virgin Islands. One thing we can do is determine how much **money** in grants is being awarded to those territories and thus being excluded from our analysis.

```
total_awarded_to_all_us_territories <-  
  grants_2000s |>  
  select(AwardOutright) |>  
  sum()  
  
total_awarded_to_non_us_territories <-  
  grants_with_no_population_info |>  
  select(AwardOutright) |>  
  sum()  
  
percent_awards_excluded <-  
  ((total_awarded_to_non_us_territories / total_awarded_to_all_us_territories) * 100) |>  
  round(2)
```

In the 2000s the NEH awarded \$952,087,551 US dollars, of which \$13,589,354 US dollars were excluded from this analysis. That exclusion represents about 1.43 percent of NEH's total awards. While that percentage is small, those awards may have had a real impact on the



residents of those regions. That potential impact isn't captured in this analysis. Ultimately, because of our exclusion, we can't say with certainty what the NEH award dollars per capita are in those regions.

### 5.3 We used the total award plus the total matching, as opposed to just the amount the NEH spent on the award.

It's possible that the NEH is more interested in direct money spent by it, as an organization, than it is by money supplied by matching. This analysis bundles matching and outright awards together into a single figure, which may distort what the NEH is looking for.

### 5.4 There are Limits to the Reproducibility of This Document

- To reproduce these findings on another machine, the user would need to get their own [Census API Token](#).
- I could have done more to make the package environment entirely reproducible. As it is, users will have to install the packages I listed in Section 2.1.
- I didn't programmatically obtain the Data Dictionary. I found it on the NEH data landing page as a PDF and downloaded it directly in a browser.

## 6 Technology Used

- A macOS laptop
- Mozilla Firefox to browse the NEH data website and download data
- Microsoft Excel and BBEdit for initial inspection of downloaded data
- git and GitHub for version control
- R and RStudio
- R libraries as detailed in Section 2.1.
- Census.gov API (used via `tidycensus`)
- [Quarto](#) for document composition and rendering

## 7 Sources

- `fct_reorder` in R package `forcats` documentation: [https://forcats.tidyverse.org/reference/fct\\_reorder.html](https://forcats.tidyverse.org/reference/fct_reorder.html)
- `as_list` in R package `xml2`: [https://xml2.r-lib.org/reference/as\\_list.html](https://xml2.r-lib.org/reference/as_list.html)
- StackOverflow: Converting a State Name to State Abbreviation: <https://stackoverflow.com/questions/5411111/converting-a-state-name-to-state-abbreviation>

- Analyzing US Census Data: Methods, Maps, and Models in R: <https://walker-data.com/census-r/an-introduction-to-tidycensus.html?q=key#getting-started-with-tidycensus>
- US Census API Key Request Form: [http://api.census.gov/data/key\\_signup.html](http://api.census.gov/data/key_signup.html)
- `usmap` package for cartographic representations of the United States