

IIJA Funding related to population

By Tony Fraser

2024-09-03

Story 1: IIJA Funding related to population

Introduction

After the Great Depression, Franklin D. Roosevelt, a New Yorker, created many jobs through public works programs such as the Works Progress Administration (WPA), the Civilian Conservation Corps (CCC), and the Public Works Administration (PWA). Around the same time, another New Yorker, Robert Moses, began rising in NYC politics. As more money became available through the New Deal, Moses expanded his team of architects, proposed more plans, and directed significant portions of New Deal funds to New York. Eventually, Robert Moses became one of the most prolific urban planners in history and one of the most powerful figures in NYC politics. In the end, close to 15% of all New Deal money was allocated to New York City and its surrounding infrastructure.

This our American history, and this assignment, beg a similar question about the Jobs Act allocation. Let's standardize this dataset and see if they align, or if we're looking at another New York City is better scenario.

Plan

1. Get data
 - Dataset 1: A list of states AND provinces
 - Dataset 2: IIJA funding (Note that Delaware is missing)
 - Dataset 3: [US Census data](#) (Plus, manually find records for missing territories)
2. Clean (column names, spaces, etc.)
3. Join states to funding to population
4. Build calculated columns for `expected_funding`, `funding_difference`, and `percentage_difference`
5. Persist a clean spreadsheet

6. Chart in Excel, save screen shots into ./story1_files/
7. Add charts into this document
8. Conclude
9. Push to github

Steps 1 - 5

We'll use python for cleaning because it works with Quarto.

```
import pandas as pd
merged_fn = './story1_files/merged_states_funding_population.csv'

funding_df = pd.read_excel('./story1_files/IIJA FUNDING AS OF MARCH 2023.xlsx')
population_df = pd.read_csv('./story1_files/NST-EST2023-ALLDATA.csv')

# We'll use this to see what's missing or incorrect.
states_list = [
    'alabama', 'alaska', 'americansamoa', 'arizona', 'arkansas', 'california',
    'colorado', 'connecticut', 'delaware', 'districtofcolumbia', 'florida',
    'georgia', 'guam', 'hawaii', 'idaho', 'illinois', 'indiana', 'iowa', 'kansas',
    'kentucky', 'louisiana', 'maine', 'maryland', 'massachusetts', 'michigan',
    'minnesota', 'mississippi', 'missouri', 'montana', 'nebraska', 'nevada',
    'newhampshire', 'newjersey', 'newmexico', 'newyork', 'northcarolina', 'northdakota',
    'northernmarianaislands', 'ohio', 'oklahoma', 'oregon', 'pennsylvania', 'puertorico',
    'rhodeisland', 'southcarolina', 'southdakota', 'tennessee', 'texas', 'utah',
    'vermont', 'virginia', 'washington', 'westvirginia', 'wisconsin', 'wyoming'
]
states_df = pd.DataFrame(states_list, columns=['state'])

funding_data = (
    pd.read_excel('./story1_files/IIJA FUNDING AS OF MARCH 2023.xlsx')
    [['State, Territory or Tribal Nation', 'Total (Billions)']]
    .rename(
        columns={'State, Territory or Tribal Nation':
                 'state', 'Total (Billions)': 'actual'})
    .assign(state=lambda df: df['state'].str.lower().str.replace(' ', ''))
)
population_data = (
    population_df[['NAME', 'POPESTIMATE2023']]
    .rename(columns={'NAME': 'state', 'POPESTIMATE2023': 'pop_2023'})
    .assign(state=lambda df: df['state'].str.lower().str.replace(' ', '')))
```

```

# step 3 - do the joins
merged_df = (
    states_df
    .merge(funding_data, on='state', how='left')
    .merge(population_data, on='state', how='left')
)

# step 4 - the calculated columns
total_population = merged_df['pop_2023'].sum()
merged_df = (
    merged_df
    .assign(
        expected=lambda df: (df['pop_2023'] / total_population) * df['actual'].sum(),
        diff=lambda df: df['actual'] - df['expected'],
        percent_diff=lambda df: (df['diff'] / df['expected']) * 100
    )
    .round({'expected': 2, 'diff': 2, 'percent_diff': 2})
)

# step 5 - persist the CSV
merged_df.to_csv(merged_fn, index=False)

# glimpse raw data
merged_df = pd.read_csv(merged_fn)
merged_df.head(10)

```