Appendix

Allyson Cui

2025-07-02

Appendix A: Version 1

Used in `Project`

Project knowledge: `appointments_2024.csv`

Prompt 1: Dataset Combination Script

You have 12 CSV files containing New Brunswick government appointment data from 2013-2024 (appointments_2013.csv through appointments_2024.csv), they are stored in raw_data. In the following analysis, always use relative path, such as filename = f"raw_data/appointments_{year}.csv".

Write a Python script in scripts/gpt4o/version1/execution1 that:

1. Loads all 12 CSV files into pandas DataFrames
2. Adds a 'year' column to each dataset before combining
3. Combines all datasets into a single DataFrame
4. Saves the combined dataset as 'step1_combined_appointments.csv' in scripts/gpt4o/version1/execution1/analysis_data
5. Prints the shape and basic info about the combined dataset

Include proper error handling and document any assumptions about data structure.

Use standard data science libraries (pandas, numpy, os, pathlib, sys) as needed.

Do not do extra steps.

Use #!/usr/bin/env python3.

Prompt 2: Key Column Extraction Script

Write a Python script in scripts/gpt4o/version1/execution1 that:

1. Loads the combined dataset from 'step1_combined_appointments.csv' in in scripts/gpt4o/version1/execution1/analysis_d
2. Identifies and extracts the key columns: "reappointed", "name", "position", "org", "year"
3. Creates a new dataset with only these key columns plus the year column
4. Saves the filtered dataset as 'step2_key_columns_data.csv' in in scripts/gpt4o/version1/execution1/analysis_data
5. Prints information about the extracted columns and any missing values

The script should be robust to minor variations in column naming.

Use standard data science libraries (pandas, numpy, os, pathlib, sys) as needed.

Do not do extra steps.

Prompt 3: Repeat Appointment Detection Script

Write a Python script in scripts/gpt4o/version1/execution1 that:

1. Loads the dataset from step 2
2. For each unique combination of “name”, “position”, and “org”:
   - Identifies all occurrences of the same person in the same position at the same organization
   - Marks all occurrences EXCEPT the first (chronologically by year) as reappointments
3. Updates the “reappointed” column for these cases
4. Handles name variations and potential duplicates intelligently
5. Prints statistics showing how many additional reappointments were identified
6. Saves the updated dataset as ‘step3_repeats_marked.csv’

Include logic to handle potential edge cases like missing dates or ambiguous name matches.

Use standard data science libraries (pandas, numpy, os, pathlib, sys) as needed.

Do not do extra steps.

Prompt 4: Employee Counting Script

Write a Python script in scripts/gpt4o/version1/execution1 that:

1. Loads the dataset from step 3
2. Groups the data by “org” (organization) and “year”
3. Counts the total number of appointments for each organization in each year
4. Creates a summary table with organizations as rows, years as columns, and counts as values
5. Handles missing values appropriately
6. Saves the counts as ‘step4_employee_counts.csv’ in scripts/gpt4o/version1/execution1/analysis_data
7. Prints the summary table and identifies organizations with the most appointments

Include validation to ensure counts are reasonable and handle any data quality issues.

Use standard data science libraries (pandas, numpy, os, pathlib, sys) as needed.

Do not do extra steps.

Prompt 5: Reappointment Counting Script

Write a Python script in scripts/gpt4o/version1/execution1 that:

1. Loads the dataset from step 3
2. Filters for records where “reappointed” is True
3. Groups by “org” and “year” and counts reappointments
4. Creates a summary table showing reappointment counts by organization and year
5. Handles cases where an organization has zero reappointments in a given year
6. Saves the reappointment counts as ‘step5_reappointment_counts.csv’ in scripts/gpt4o/version1/execution1/analysis_data
7. Prints summary statistics about reappointment patterns

Include debugging output to show sample grouped data before creating the pivot table. Ensure year columns are handled as integers (not floats) to match the employee counts format. Add validation to verify the pivot table contains actual counts rather than zeros.

Use standard data science libraries (pandas, numpy, os, pathlib, sys) as needed.

Do not do extra steps.

Prompt 6: Rate Calculation Script

Write a Python script in scripts/gpt4o/version1/execution1 that:

1. Loads both the employee counts (step 4) and reappointment counts (step 5)
2. Merges the two datasets appropriately
3. Calculates the reappointment rate as: reappointments / total_employees for each org/year combination
4. Handles division by zero and missing data cases
5. Creates a comprehensive table with columns: org, year, total_employees, reappointments, reappointment_rate
6. Saves the results as 'step6_reappointment_rates.csv' in scripts/gpt4o/version1/execution1/analysis_data
7. Prints organizations with the highest average reappointment rates

Include proper handling of edge cases and data validation.

Use standard data science libraries (pandas, numpy, os, pathlib, sys) as needed.

Do not do extra steps.

Prompt 7: Maximum Rate Identification Script

Write a Python script in scripts/gpt4o/version1/execution1 that:

1. Loads the reappointment rates from step 6
2. For each year, identifies the organization with the highest reappointment rate
3. Handles ties appropriately (document your approach)
4. Creates a time series showing the top organization and its rate for each year
5. Saves results as 'step7_yearly_max_rates.csv' with columns: year, top_org, max_rate in scripts/gpt4o/version1/execution1/analysis_data
6. Creates a visualization showing the trend over time to be stored as "step7_yearly_max_rates.png".
7. Prints the year-by-year results

Include logic to handle edge cases like ties or missing data for certain years.

Use standard data science libraries (pandas, numpy, os, pathlib, sys) as needed.

Do not do extra steps.

Prompt 8: Annual Government-Wide Proportion Script

Write a Python script in scripts/gpt4o/version1/execution1 that:

1. Loads the dataset from step 3 (with reappointment flags)
2. For each year (2013-2024), calculates the overall proportion of reappointments across the entire government:

- Count total appointments across all organizations for the year
- Count total reappointments across all organizations for the year
- Calculate proportion = total_reappointments / total_appointments

3. Creates a time series with columns: year, total_appointments, total_reappointments, reappointment_proportion
4. Saves the results as ‘step8_annual_proportions.csv’
5. Creates a visualization showing the government-wide reappointment proportion trend over time to be stored as “step8_annual_reappointment_proportions.png”
6. Prints the year-by-year proportions

This calculates the overall reappointment rate for the entire New Brunswick government each year, not by organization.

Use standard data science libraries (pandas, numpy, os, pathlib, sys) as needed.

Do not do extra steps.

Prompt 9: Trend Analysis Script

Write a Python script in scripts/gpt4o/version1/execution1 that:

1. Loads the annual government-wide proportions from step 8
2. Prepares the data for regression analysis (year as X, government-wide reappointment proportion as Y)
3. Fits a linear regression model using scipy.stats.linregress
4. Calculates comprehensive statistics including slope, intercept, R-squared, p-value, standard error, and 95% confidence intervals
5. Performs regression diagnostics including Durbin-Watson test and outlier detection
6. Tests whether the trend coefficient is statistically significant ($p < 0.05$)
7. Determines if the trend is increasing (positive coefficient) or decreasing (negative coefficient)
8. Calculates annual change in percentage points and total change over the 12-year period
9. Saves detailed statistical results including data summary, regression equation, and conclusions to ‘step9_regression_results.txt’
10. Prints the final answer: Is the government-wide reappointment proportion trend increasing or declining over the 12-year period, and is it statistically significant?

Include comprehensive error handling and clear documentation throughout the code.

Use standard data science libraries (pandas, numpy, os, pathlib, sys) as needed.

Do not do extra steps.

Appendix B: Version 2

Used in `Project`

Project knowledge: `appointments_2024.csv`, `Code Generating Guidance`

Code Generating Guideline

Dataset and Environment Context:

- You have 12 CSV files containing New Brunswick government appointment data from 2013-2024, however, I am only providing appointments_2024.csv as an example of the entire dataset, you may assume the other csv files exist in my local folder, but you will not have it here
- File naming convention: appointments_2013.csv, appointments_2014.csv, . . . , appointments_2024.csv
- Files are located in: raw_data
- Save all intermediate results to: scripts/gpt4o/version2/execution1/analysis_data/
- Use this exact file naming convention for outputs:
  - step1_combined_appointments.csv
  - step2_key_columns_data.csv
  - step3_repeats_marked.csv
  - step4_employee_counts.csv
  - step5_reappointment_counts.csv
  - step6_reappointment_rates.csv
  - step7_yearly_max_rates.csv
  - step7_yearly_max_reappointment_rates.png
  - step8_annual_proportions.csv
  - step8_annual_reappointment_proportions.png
  - step9_regression_results.txt

Required Python Libraries:

- pandas for data manipulation
- numpy for calculations
- scipy.stats for regression analysis
- pathlib for file handling
- Standard error handling and validation practices

Data Structure Expectations:

- Each CSV contains columns that may include: name, position, organization/org, reappointed, dates, etc.
- The “reappointed” column may be boolean (True/False) or text values
- Handle missing values and data quality issues appropriately

Code Requirements:

- Include comprehensive error handling and file existence checking
- Print progress updates and validation statistics at each step
- Document all analytical decisions and assumptions in comments
- Create output directories if they don’t exist
- Validate data integrity between steps

Research Question:

Which government branch in New Brunswick most frequently reappoints past appointees, and is this trend increasing or declining, over the past 12 years?

Prompt 1:

Use code generating guideline, write a Python script in scripts/gpt4o/version2/execution1 for step 1:

Combine the 12 raw datasets.

Prompt 2:

Use code generating guideline, write a Python script in scripts/gpt4o/version2/execution1 for step 2:

Extract and retain the key columns: “reappointed”, “name”, “position”, “org”, and “year”.

Prompt 3:

Use code generating guideline, write a Python script in scripts/gpt4o/version2/execution1 for step 3:

Mark “reappointed” as true for repeated “name”-“position”-“org” combinations except for the first appearance.

Prompt 4:

Use code generating guideline, write a Python script in scripts/gpt4o/version2/execution1 for step 4:

Count the total number of employees for each “org” in each year.

Prompt 5:

Use code generating guideline, write a Python script in scripts/gpt4o/version2/execution1 for step 5:

Count how many times each “org” appears with “reappointed” marked as true for each year.

Prompt 6:

Use code generating guideline, write a Python script in scripts/gpt4o/version2/execution1 for step 6:

Calculate the reappointment rate as reappointments divided by total employees for each org-year pair.

Prompt 7:

Use code generating guideline, write a Python script in scripts/gpt4o/version2/execution1 for step 7:

Identify the organization with the highest reappointment rate for each year.

Prompt 8:

Use code generating guideline, write a Python script in scripts/gpt4o/version2/execution1 for step 8:

Compute the government-wide reappointment proportion for each year.

Prompt 9:

Use code generating guideline, write a Python script in scripts/gpt4o/version2/execution1 for step 9:

Run a linear regression on the annual reappointment proportions to assess trend direction and significance.

Appendix C: Version 3

Used in `Project`

Project knowledge: `appointments_2024.csv` and `Code Generating Guideline`

Code Generating Guideline

Dataset and Environment Context:

- You have 12 CSV files containing New Brunswick government appointment data from 2013-2024, however, I am only providing appointments_2024.csv as an example of the entire dataset, you may assume the other csv files exist in my local folder, but you will not have it here
- File naming convention: appointments_2013.csv, appointments_2014.csv, . . . , appointments_2024.csv
- Files are located in: raw_data
- Save all intermediate results to: scripts/gpt4o/version2/execution1/analysis_data/
- Use this exact file naming convention for outputs:
  - step1_combined_appointments.csv
  - step2_key_columns_data.csv
  - step3_repeats_marked.csv
  - step4_employee_counts.csv
  - step5_reappointment_counts.csv
  - step6_reappointment_rates.csv
  - step7_yearly_max_rates.csv
  - step7_yearly_max_reappointment_rates.png
  - step8_annual_proportions.csv
  - step8_annual_reappointment_proportions.png
  - step9_regression_results.txt

Required Python Libraries:

- pandas for data manipulation
- numpy for calculations
- scipy.stats for regression analysis
- pathlib for file handling
- Standard error handling and validation practices

Data Structure Expectations:

- Each CSV contains columns that may include: name, position, organization/org, reappointed, dates, etc.
- The "reappointed" column may be boolean (True/False) or text values
- Handle missing values and data quality issues appropriately

Code Requirements:

- Include comprehensive error handling and file existence checking
- Print progress updates and validation statistics at each step

- Document all analytical decisions and assumptions in comments
- Create output directories if they don't exist
- Validate data integrity between steps

Research Question:

Which government branch in New Brunswick most frequently reappoints past appointees, and is this trend increasing or declining, over the past 12 years?

Prompt:

Which government branch in New Brunswick most frequently reappoints past appointees, and is this trend increasing or declining over the past 12 years?

Use code generating guideline, write a Python script in scripts/gpt4o/version3/execution1 that does:

- Combine the 12 raw datasets.
- Extract and retain the key columns: "reappointed", "name", "position", "org", and "year".
- Mark "reappointed" as true for repeated "name"-"position"-"org" combinations except for the first appearance.
- Count the total number of employees for each "org" in each year.
- Count how many times each "org" appears with "reappointed" marked as true for each year.
- Calculate the reappointment rate as reappointments divided by total employees for each org-year pair.
- Identify the organization with the highest reappointment rate for each year.
- Compute the government-wide reappointment proportion for each year.
- Run a linear regression on the annual reappointment proportions to assess trend direction and significance.