APIs and Election Forecasting

Conceptual Questions

Please write three to ten sentence explanations for each of the following questions. **You are only required to answer ONE of the two questions below.**

1. In the social sciences, what are two advantages of collecting data via an API (instead of web scraping)? What are two limitations or risks (e.g., coverage bias, changing endpoints, rate limits, versioning, missingness)? Explain how you would document API-based data provenance so another researcher could replicate your dataset later.
2. Why is API key management part of doing professional, reproducible data science? Explain (i) what can go wrong if keys are hard-coded and pushed to GitHub, and (ii) one concrete approach for storing keys locally (e.g., environment variables) while keeping your analysis reproducible for yourself and collaborators.

Applied Exercises

Use the code in the week's code tutorial and the lecture slides to answer the following questions.

3. **Create an API key and connect to the API.** This week you will use the `fredr` package, which requires a FRED API key.
   - Create a FRED API key.
   - Update the provided script so it reads your key from a local environment variable (recommended) rather than hard-coding it in the file.
   - Confirm your API call works by successfully downloading at least one FRED time series (e.g., unemployment, GDP, or CPI) for election years.
4. **Run the baseline forecaster (replicate).** Using the provided code (`api_build_model.r`), get the baseline pipeline running end-to-end on your machine.
   - If any file paths are absolute, modify them to use a **relative path** that will work inside your course project folder.
   - Produce at least one baseline output showing model predictions on held-out data (e.g., 2020), such as a printed table, summary statistics, or a map.
5. **Build a better out-of-sample forecaster.** Starting from the baseline, improve out-of-sample performance for election results that are not used to fit the model.
   - Define an explicit out-of-sample design (choose one):
     (a) Hold out the most recent election year in the dataset (e.g., train on years < 2020, test on 2020), *or*
     (b) Do a "leave-one-election-out" evaluation (train on all but one election year; repeat).
   - Implement **at least two** model improvements. Examples include:
     - adding additional FRED indicators (via the API) and justifying them,

  - changing the functional form (interactions, nonlinear terms),
  - using regularization (ridge/lasso/elastic net),
  - switching model families (e.g., random forest, gradient boosting),
  - improving feature engineering (e.g., transforms, per-capita changes),
  - improving how you aggregate to a prediction target (national vs state).
- Report performance **out-of-sample** using at least one quantitative metric (e.g., MAE, RMSE) and compare it to the baseline.

6. **Communicate the comparison.** Provide:
   - one table that compares baseline vs improved performance (out-of-sample), and
   - one figure that communicates model fit or errors (e.g., predicted vs actual; error by state; time series comparison).
7. **Challenge Question (Optional — if you finish early):**
   (a) **Forecast validation and "small n" in elections.** Defend a position (8–12 sentences): Do we have enough realized election outcomes to validate a national election forecaster? Discuss the trade-offs of treating forecasting as (i) national outcomes (one datapoint per election) versus (ii) state-level outcomes (many datapoints per election). Explain what you would report as evidence that your model is "working" and what would still remain uncertain.

   (b) **API engineering + reproducibility challenge.** Write a small wrapper function `get_fred_features(series_ids, years)` that:
   - downloads multiple FRED series via the API,
   - caches the raw results to disk (so repeated runs do not re-download),
   - gracefully handles API failures (e.g., `tryCatch` + informative messages).

   Demonstrate it by adding **one new** economic series to your model and showing whether it helps out-of-sample performance.