Homework 5

Problem 1 - Iron Bank

The Securities and Exchange Commission (SEC) is investigating the Iron Bank, where a cluster of employees have recently been identified in various suspicious patterns of securities trading that violate federal "insider trading" laws. Here are few basic facts about the situation:

- Of the last 2021 trades by Iron Bank employees, 70 were flagged by the SEC's detection algorithm.
- But trades can be flagged every now and again even when no illegal market activity has taken place. In fact, the SEC estimates that the baseline probability that any *legal* trade will be flagged by their algorithm is 2.4%.
- For that reason, the SEC often monitors individual and institutional trading but does not investigate incidents that look plausibly consistent with random variability in trading patterns. In other words, they won't investigate unless it seems clear that a cluster of trades is being flagged at a rate significantly higher than the baseline rate of 2.4%.

Are the observed data (70 flagged trades out of 2021) consistent with the SEC's null hypothesis that, over the long run, securities trades from the Iron Bank are flagged at the same 2.4% baseline rate as that of other traders?

Use Monte Carlo simulation (with at least 100000 simulations) to calculate a p-value under this null hypothesis. Include the following items in your write-up:

- the null hypothesis that your are testing;
- the test statistic you used to measure evidence against the null hypothesis;
- a plot of the probability distribution of the test statistic, assuming that the null hypothesis is true;
- the p-value itself;
- and a one-sentence conclusion about the extent to which you think the null hypothesis looks plausible in light of the data. This one is open to interpretation! Make sure to defend your conclusion.

Problem 2: health inspections

The local Health Department is investigating a popular local restaurant chain, Gourmet Bites, after receiving a higher-than-usual number of health code violation reports. Here are a few key points about the situation:

- Over the last year, 1500 health inspections were conducted across various restaurants in the city, with various branches of Gourmet Bites inspected a total of 50 times.
- Of these 50 inspections, 8 resulted in health code violations being reported.
- Typically, the Health Department's data shows that, on average, 3% of all restaurant inspections result in health code violations due to random issues that can occur even in well-managed establishments.

The Health Department wants to ensure that any action taken is based on solid evidence that Gourmet Bites' rate of health code violations is significantly higher than the citywide average of 3%.

Question: Are the observed data for Gourmet Bites consistent with the Health Department's null hypothesis that, on average, restaurants in the city are cited for health code violations at the same 3% baseline rate?

Use a Monte Carlo simulation (with at least 100,000 simulations) to calculate a p-value under this null hypothesis. Follow the same answer format as in the prior problem.

Problem 3: LLM watermarking

Watermarking output from a large language model (LLM) like ChatGPT involves embedding a unique, identifiable pattern or marker into the generated text. This marker is designed to be difficult to detect or remove without significantly altering the content or meaning of the text. The primary purpose of watermarking is to enable the tracing of the origin of the content, ensuring accountability and helping to prevent misuse of the technology. For example, watermarking can help identify when text has been generated by an AI model, distinguishing it from human-generated content.

The general principle of watermarking involves altering the output of the LLM in a subtle, consistent way that does not compromise the readability or coherence of the text. This alteration must be detectable through analysis, allowing the identification of the source of the content. The challenge lies in designing a watermark that is robust enough to withstand attempts at removal or obfuscation while remaining inconspicuous to casual observation. This is an active research area in statistics, machine learning, and AI.

In this problem, we'll consider a really simple example of watermarking, where we imagine than an LLM manipulating the frequency of certain letters in the generated sentences to deviate from their typical distribution in English. For instance, the model might be programmed to use the letter "z" more frequently than usual, or when given the choice of two equally good synonyms, to choose the one with rarer letters. This would create a distinctive pattern in the text that could be identified through statistical analysis, serving as a watermark. (Real watermarks are more sophisticated than this, but this simple "frequency shift" serves to convey the general idea.) The key is to ensure that these alterations do not significantly impact the naturalness or readability of the text, allowing the watermark to remain hidden in plain sight.

Here you'll build on the use of the chi-squared goodness of fit statistic that we calculated in solving Caesar ciphers in class. Refer to that analysis for necessary building blocks in your code. You'll also want the "letter_frequencies.csv" data set of letter frequencies that we calculated from Project Gutenberg texts.

The fundamental goal of this question is to look at ten sentences and test them for the presence of a watermark. Nine are "normal" sentences, drawn from actual English text. One was generated by an LLM with a watermark based on the frequency shift idea described above. Can you tell which sentence it is?

Let's begin!

Part A: the null or reference distribution

To start with, download the data in "brown_sentences.txt" from Canvas. This file contains a collection of English sentences extracted from the Brown Corpus. The Brown Corpus is a well-known and widely used text corpus in linguistics and natural language processing (NLP). It was compiled in the 1960s at Brown University and consists of over one million words of American English text, drawn from a diverse range of sources, including newspapers, books, and government documents. The sentences in brown_sentences.txt represent a sample of this corpus and provide a snapshot of English sentences usage across different genres and contexts. The file has one sentence per line.

The point of this data set is allow you to calculate a "reference" or null distribution of the chi-squared test statistic based on letter frequency. Essentially, you'll be answering the question: what does the chi-squared statistic look like across lots of normal English sentences not generated by an LLM? You should follow these steps:

1. Read the sentences: Load the sentences from brown_sentences.txt into your R environment. Look into the `readLines` function, which should be useful here (although not the only way).

2. Preprocess the text: For each sentence, remove non-letter characters, convert the text to uppercase, and count the occurrences of each letter. (We did this in our Caesar cipher example; re-use that code as appropriate.)

3. Calculate letter count: For each sentence, calculate the frequency of each letter. This will give you the observed letter counts for each sentence.

4. Compare with expected count: Using our predefined letter frequency distribution for English (i.e. the one we've used previously), calculate the expected count of each letter in each sentence based on the sentence length.
5. Compute the chi-squared statistic: For each sentence, calculate the chi-squared statistic to measure the discrepancy between the observed and expected counts of each letter.

6. Compile the distribution: Collect the chi-squared statistics from all sentences to form your reference or null distribution. This distribution represents the range of chi-squared values you might expect to see in normal English sentences based on the predefined letter frequency distribution.

By creating this null distribution, you will be able to address Part B. Consider a for loop or R's `sapply` function to calculate chi-squared for each sentence in the data.

Part B: checking for a watermark

OK, so you now know what chi-squared looks like across thousands of normal English sentences. Now consider the following ten sentences:

1. She opened the book and started to read the first chapter, eagerly anticipating what might come next.
2. Despite the heavy rain, they decided to go for a long walk in the park, crossing the main avenue by the fountain in the center.
3. The museum's new exhibit features ancient artifacts from various civilizations around the world.
4. He carefully examined the document, looking for any clues that might help solve the mystery.
5. The students gathered in the auditorium to listen to the guest speaker's inspiring lecture.
6. Feeling vexed after an arduous and zany day at work, she hoped for a peaceful and quiet evening at home, cozying up after a quick dinner with some TV, or maybe a book on her upcoming visit to Auckland.
7. The chef demonstrated how to prepare a delicious meal using only locally sourced ingredients, focusing mainly on some excellent dinner recipes from Spain.
8. They watched the sunset from the hilltop, marveling at the beautiful array of colors in the sky.
9. The committee reviewed the proposal and provided many points of useful feedback to improve the project's effectiveness.
10. Despite the challenges faced during the project, the team worked tirelessly to ensure its successful completion, resulting in a product that exceeded everyone's expectations.

For each of these sentences, calculate a p-value under the null hypothesis that the sentence follows the "typical" English letter distribution. Use chi-squared based on letter frequencies as a test statistic. You will need the null or reference distribution you calculated in Part A. Show a table of these p-values to three decimal places.

One of these sentences has been produced by an LLM, but watermarked by asking the LLM to subtly adjust its frequency distribution over letters. Which sentence is it? How do you know?

Note: I recommend the following: "Hi ChatGPT! Can you take the following ten sentences and put them into an R vector for me so I can loop over the sentences in my R script? (then paste the ten sentences...)"