Homework 3

Assignment scoring:

Section	Required Files	Score
Reddit Weekends	`reddit_weekends.py`	35
+ figures, short answers	`reddit_weekends.ipynb`	10
Chess Ratings	`chess_ratings.py`	35
+ figures, short answers	`chess_ratings.ipynb`	20

In general your homework will be autograded, i.e. you must **not modify the signature of the defined functions** (same inputs and outputs). For questions involving Numpy and Pandas, you **must** handle any iteration through calls to relevant functions in the libraries. This means you may not use any native Python for loops, while loops, list comprehension, etc. when something could have been done with a native pandas/numpy call. Loops are otherwise permitted.

Submitting

To submit the files, please submit **only the required files** (listed in the above table) that you completed to **gradescope**; do not include data or other miscellaneous files. You do not need to submit all of the files at once in order to run the autograder. For example, if you only completed `reddit_weekends.py`, you don't need to submit the rest of the files in order to get feedback for `reddit_weekends.py`.

1. Reddit Weekends

This question uses data derived from the Reddit Comment archive, which is a collection of every Reddit comment, distributed as 150 GB of compressed JSON.

The provided file `reddit-counts.json.gz` contains a count of the number of comments posted daily in each Canadian-province subreddit, and in /r/canada itself. (The values will differ slightly from The Truth: the timezones may not be set correctly, so there will be some comments categorized incorrectly around midnight: I'm willing to live with that.) Again, the format is gzipped line-by-line JSON. It turns out Pandas (≥0.21) can handle the compression automatically and we don't need to explicitly uncompress: ``

```
counts = pd.read_json(sys.argv[1], lines=True)
```

The question at hand: are there a different number of Reddit comments posted on weekdays than on weekends?

For this question, we will look only at values (1) in 2012 and 2013, and (2) in the /r/canada subreddit. Start by creating a DataFrame with the provided data, and separate the weekdays from the weekends.

Hint: check for `datetime.date.weekday` either 5 or 6.

Complete the program `reddit_weekends.py` for this question, and follow along the `reddit_weekends.ipynb` notebook. You will be assigned marks for your implementation in the python file, and for the figures and short answers in the notebook. The python file also has code setup to run for you; you can take the input data file via the command line:

```
python3 reddit_weekends.py ./data/reddit-counts.json.gz
```

Note that the output produced by the above command will not be marked; we will be testing your functions directly via unit tests as usual.

1.1 Student's T-Test

- Complete `reddit_weekends.py:process_data()`
- Complete `reddit_weekends.py:tests()`

Use scipy.stats to do a T-test on the data to get a p-value. Can you conclude that there are a different number of comments on weekdays compared to weekends? Try `stats.normaltest` to see if the data is normally-distributed, and `stats.levene` to see if the two data sets have equal variances. Now do you think you can draw a conclusion? (Hint: no. Just to check that we're on the same page: I see a “0.0438” here.)

1.2 Fix 1: transforming data might save us.

- Complete the figures in `reddit_weekends.ipynb`

Have a look at a histogram of the data. You will notice that it's skewed: that's the reason it wasn't normally-distributed in the last part. Transform the counts so the data doesn't fail the normality test. Likely options for transforms: np.log, np.exp, np.sqrt, counts**2. Pick the one of these that comes closest to normal distributions. [Unless I missed something, none of them will pass the normality test. The best I can get: one variable with normality problems, one okay; no equal-variance problems.]

1.3 Fix 2: the Central Limit Theorem might save us.

- Complete `reddit_weekends.py:central_limit_theorem()`
- Complete the figures in `reddit_weekends.ipynb`

The central limit theorem says that if our numbers are large enough, and we look at sample means, then the result should be normal. Let's try that: we will combine all weekdays and weekend days from each year/week pair and take the mean of their (non-transformed) counts.

Hints: you can get a “year” and “week number” from the first two values returned by date.isocalendar(). This year and week number will give you an identifier for the week. Use Pandas to group by that value, and aggregate taking the mean. Note: the year returned by isocalendar isn't always the same as the date's year (around the new year). Use the year from isocalendar, which is correct for this.

Check these values for normality and equal variance. Apply a T-test if it makes sense to do so. (Hint: yay!)

We should note that we're subtly changing the question here. It's now something like “do the number of comments on weekends and weekdays for each week differ?”

1.4 Fix 3: a non-parametric test might save us.

- Complete `reddit_weekends.py:mann_whitney_u_test()`

The other option we have in our toolkit: a statistical test that doesn't care about the shape of its input as much. The Mann–Whitney U-test does not assume normally-distributed values, or equal variance.

Perform a U-test on the (original non-transformed, non-aggregated) counts. Note that we should do a two-sided test here, which will match the other analyses. Make sure you get the arguments to the function correct.

Again, note that we're subtly changing the question again. If we reach a conclusion because of a U test, it's something like “it's not equally-likely that the larger number of comments occur on weekends vs weekdays.”

1.5 Questions

Answer these questions in the section specified in `reddit_weekends.ipynb`:

- Which of the four transforms suggested got you the closest to satisfying the assumptions of a T-test?
- I gave imprecise English translations of what the by-week test, and the Mann-Whitney test were actually testing. Do the same for the original T-test, and for the transformed data T-test. That is, describe what the conclusion would be if you could reject the null hypothesis in those tests.
- Of the four approaches, which do you think actually does a better job of getting an answer for the original question: “are there a different number of Reddit comments posted on weekdays than on weekends?” Briefly explain why. (It's not clear to me that there is a single correct answer to this question.)
- When are more Reddit comments posted in /r/canada, on average weekdays or weekends?

2. Chess Ratings

When comparing the performance between two groups, a common mistake is to compare them based on the top entities in each group, especially in scenarios in which one of the groups has a significantly larger number of samples to the other (overrepresented). One such example is in chess, where people have incorrectly tried to argue that female players are worse than male players because when you consider the top players, no female has ever been a world champion, or the Elo (a method for calculating the relative skill levels of players) difference between the best female and male is large, etc.

Why is this a mistake? Simply put, this neglects the participation of these groups. Naively, if two groups are drawn from the same distribution, the overrepresented group will have "more shots" at the tails of the distribution. For example, say there are two groups of people: A and B. Group A has 10 people, group B has 2. Each of the 12 people gets randomly assigned a number between 1 and 100 (with replacement). Then, take the max in Group A as the score for Group A and the max in Group B as the score for Group B. On average, Group A will score ~91 and Group B ~67, where the only difference between the two groups is the size. The larger group has more shots at a high score, so will on average get a higher score. The fair way to compare these unequally sized groups is by comparing their means (averages), not their top values. Of course, in this example, that would be 50 for both groups – no difference! Note that this only highlights the statistical effect of simply comparing overrepresented and underrepresented groups and neglects any potential issues or other biases in the data.

In this section, we will explore this in the context of the aforementioned chess ratings. No steering code is provided in the python file; instead you can follow along in the `chess_ratings.ipynb` notebook.

2.1 Load and Clean data

- Complete `chess_ratings.py:parse_xml()`
- Complete `chess_ratings.py:clean_data()`

You are provided with a (zipped) XML file:

```
./data/standard_oct22frl_xml.zip
```

This was obtained from the FIDE player database which can be found here. Note that the ratings are updated fairly regularly, so we are using the fixed timestamp of 06 Oct 2022, which is provided for you in the data directory. You should not need to download anything.

Complete the `parse_xml()` method; note that the first part of the method has some code to automatically extract the zipfile for you. Make sure you obtain the following data from the XML (and use them as the column names) in your dataframe:

```
["name", "rating", "sex", "birthday", "country", "flag", "title"]
```

Next, complete `clean_data()`: drop players with NaN birthdays, convert the numeric types into the appropriate type, and filter for birthdays `<=2002` (as Elo scores for people < 20 years old can be unreliable).

2.2 Compare rating distributions

- Complete `chess_ratings.py:bin_counts()`
- Complete the figures in `chess_ratings.ipynb`

Now let's compare the distribution of ratings for male and female players. Since the data is quite fine-grained, we'll need to bin the ratings. Complete the `bin_counts()` data, which should handle binning for arbitrary data and choice of bins. In addition to returning the raw counts, also return the normalized counts (`count_norm`).

Next, you will use this method to complete the figures in `chess_ratings.ipynb`. For the figures, chose a bin *width* of 50, a range from 1000 - 2900 (make sure your range extends to 2900!), and the bin *centers* as the middle point of each bin (i.e. for a bin between [1500, 1550), the center would be 1525.). Note that when defining your bins, you will find that `len(bin_centers) = len(bins) - 1`

Using the binned data, draw two lineplots of the binned data side by side; one containing the raw counts (`"count"`), and the other containing the normalized counts (`"count_norm"`), and M/F should be two different colours. Make sure to include these figures in your notebook when you submit.

2.3 Permutation Tests

- Complete `chess_ratings.py:PermutationTests.job()`
- Complete `chess_ratings.py:sample_two_groups()`

Finally, we'll conduct the permutation tests as outlined in the thought experiment of the introduction. Take the full cleaned dataset (both male and females), and randomly sample two groups without replacement (i.e. shuffle the players). The size of the groups should reflect the real world difference we wish to study, i.e. the size of the male and female group. Complete `PermutationTests.job()`, which implements the sampling part of this experiment, and returns the maximum value of the over and underrepresented groups respectively.

Then, complete the `sample_two_groups()` method, which runs this experiment `n_iter` times. Once completed, run this experiment in the notebook with at least `n_iter=1000`. Run the cell which prints the mean difference obtained from the permutation tests, as well as the real differences. Make sure to include these printed results in your notebook when you submit.

2.4 Questions

Answer these questions (1-3 sentences each) in the section specified in `chess_ratings.ipynb`:

1. Interpret the results - can you come to any conclusion? Recall that claim discussed in the introduction to this question was "men are better than woman at chess because most of the top players are men". (*Note: presumably part of your answer here will involve your answer to the next question.*)
2. Do you think the numbers obtained here tell the whole story? What might be some issues with the analysis conducted here? Is the data we are working with biased in any way (other than an overrepresentation bias)? Is ELO a good metric, and can it be used to answer the original question? Are there differences in the social, cultural, systemic treatment of men and women which may prevent the underrepresented group from achieving similar results? Anything else?

The point of these questions is to highlight that data is a limited representation of the real world. It is critical for us as data scientists to take a step back when looking at a result and think about how it connects to the real world, rather than just naively assuming that the data and experimental setup is good which often results in flawed/incorrect conclusions. There could be multiple causal factors that determines the relationship that are independent to the original hypothesis: using data that doesn't really reflect the hypothesis you want to test, biased data (including overrepresented groups), real-world systemic differences between groups, etc.

References

- **Reddit Weekends** is based off of Greg Baker's data science course at SFU.
- **Chess Ratings** is based off of the analysis in this blog post.