Name: Faye(Lifei) Wang

Date: 2/6/2025

Part 1: Getting to know your data (5 Points)

- **t1_users_active_mins.csv**

This shows how much time each user actually spent on the site daily, aka active_mins, after we rolled out the experiment. Basically, it's their activity minutes logged by date once the test was live.

- **t2_users_variant.csv**

This tells us which bucket each user was in during the test — either the control group (no changes) or the treatment group (saw the new feature). Also notes when they signed up.

- **t3_users_active_mins_pre.csv**

This is their pre-test baseline — how many minutes people were normally spending on the site before we started messing with anything. Good for comparing "before vs after".

- **t4_users_attributes.csv**

The user profile stuff — like whether they're casual readers, active contributors, etc., plus basic demographics like gender. Helps slice the data later.

- **table_schema.txt**

The cheat sheet explaining all the column names and what each field actually means.

Part 2: Organizing the Data (15 Points)

- **1. What is the overall objective of this study?**

Basically, we're trying to see if the new website design actually makes people stick around longer, cuz sometimes more time spent = better engagement

- **2. What data do we need to reach that objective?**

Two key files:

t1_users_active_mins.csv — Shows actual daily minutes people spent after we launched the redesign.

t2_users_variant.csv — Tells us who was in the test group (saw the new design) vs. control group (old design).

- **3. How is the data in t1 currently organized?**

Right now, t1 is just a giant list of daily time logs — like:
"User ID #123 spent 45 mins on Jan 1, 22 mins on Jan 2..."
But we don't know which users were actually testing the new design.
Also, we have to add items up to get the total minutes per user spent per day on the website.

- **4. How should the data in t1 be organized to be useful?**

We need to merge t1 and t2 using the user ID (uid):

(1) Grabbed the variant_number (0=control, 1=treatment) from t2

(2) Slapped that label onto every row in t1

(3) Ended up with a supercharged table showing:

Who's in which group

Exactly how long they stayed each day

The date they were active

(4) Also, we have to add items up to get the total minutes per user spent per day on the website.

- **5. Organize it.**

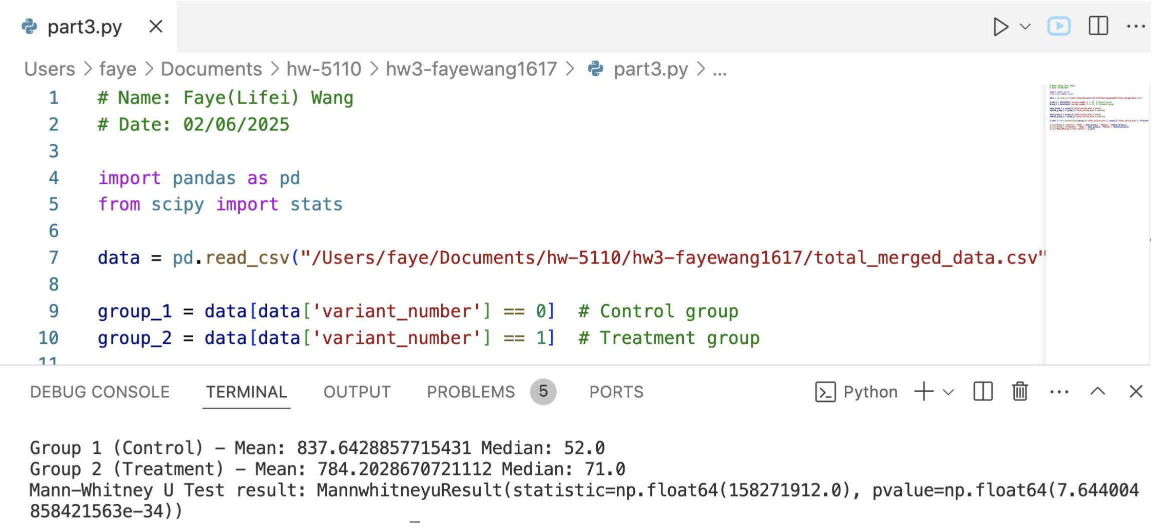
My output is total_merged_data.csv, and the code document is part2.py

Part 3: Statistical Analysis (10 Points)

- **1. Is there a statically difference between group 1 and group 2?**

Yes. The data showed distinct differences in mean and median values between the two groups (Control group: 837.64 vs 52.0 minutes; Treatment group: 784.20 vs 71.0 minutes), reflecting a highly skewed distribution. Given this skewness, the Mann-Whitney U test was used instead of a traditional t-test.

The U-test results revealed a very large U-statistic (158,271,912.0) and a p-value effectively close to zero (7.64e-34), providing strong evidence of a statistically significant difference between the control and treatment groups.


part3.py
Users > faye > Documents > hw-5110 > hw3-fayewang1617 > part3.py > ...
DEBUG CONSOLE
TERMINAL
OUTPUT
PROBLEMS 5
PORTS
Python

```python
# Name: Faye(Lifei) Wang
# Date: 02/06/2025

import pandas as pd
from scipy import stats

data = pd.read_csv("/Users/faye/Documents/hw-5110/hw3-fayewang1617/total_merged_data.csv"

group_1 = data[data['variant_number'] == 0]  # Control group
group_2 = data[data['variant_number'] == 1]  # Treatment group
```

```
Group 1 (Control) - Mean: 837.6428857715431 Median: 52.0
Group 2 (Treatment) - Mean: 784.2028670721112 Median: 71.0
Mann-Whitney U Test result: MannwhitneyuResult(statistic=np.float64(158271912.0), pvalue=np.float64(7.644004
858421563e-34))
```

- **2. What is the mean and median for group 1 and group 2?**

As shown below,

the mean for group 1 is 837.64 minutes, and the median for group 1 is 52.0 minutes.

the mean for group 2 is 784.20 minutes, and the median for group 2 is 71.0 minutes.

- **3. What can you conclude based on that data?**

The U-test results (U-statistic = 158,271,912.0; p-value = 7.64e-34) confirmed a statistically significant difference between the two groups.

This indicates that users in the treatment group (group 2) spent considerably more time on the site compared to the control group (group 1). The data indicates that the new design influenced user engagement, but further analysis may be needed to fully understand the distributional differences.

Part 4: Digging a Little Deeper (25 Points)

- **1. Can you trust that the results? Why or why not?**

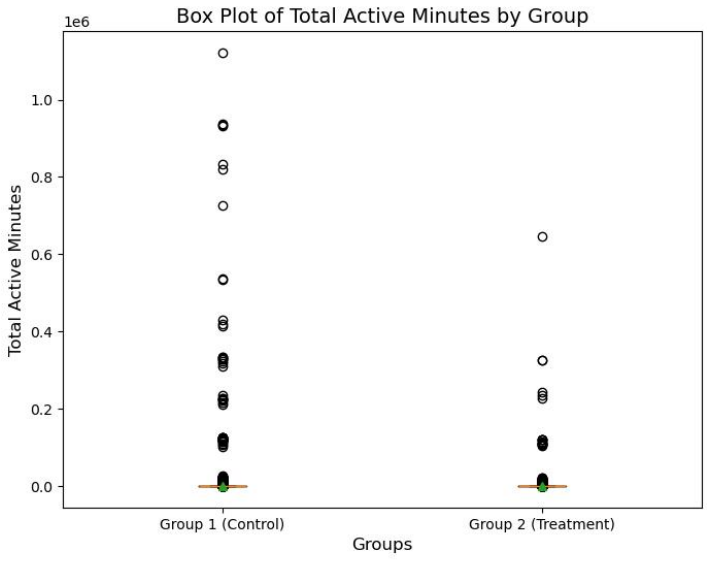
Not really. Although U-test results do show a significant difference between the two groups, which suggests that the new platform design has an impact. However, the noticeable gap between the mean and median in both groups indicates a heavily skewed data distribution, which could potentially affect the reliability of the conclusion.

Another issue is that the data isn't normally distributed, and there might be **outliers** in the dataset. These outliers could have a significant impact on the analysis as well, skewing the results.

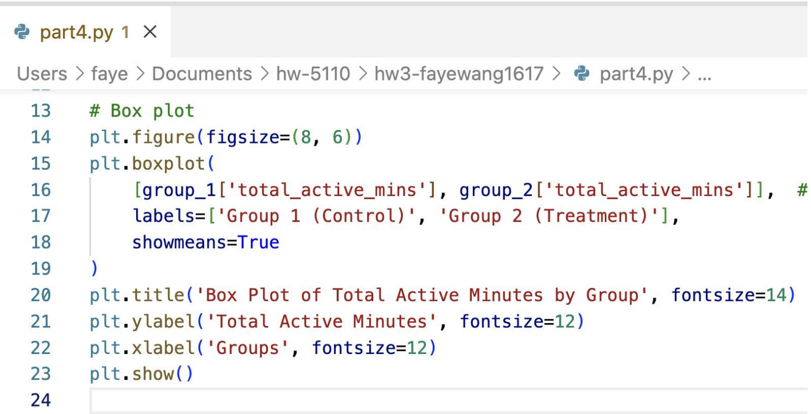
- **2. Is the data normally distributed?**

The data isn't normally distributed. We can tell from the big gap between the mean and median—837.64 vs 52.0 for group 1, and 784.20 vs 71.0 for group 2. It's a clear sign of a heavily skewed distribution.

- **3. Plot a box plot of group 1 and group 2.**



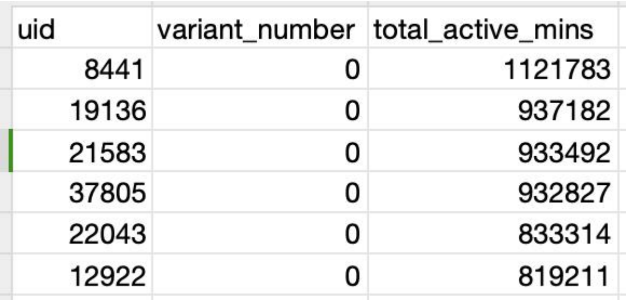
Here is the code:


part4.py 1 ×
Users > faye > Documents > hw-5110 > hw3-fayewang1617 > part4.py > ...

```python
# Box plot
plt.figure(figsize=(8, 6))
plt.boxplot(
    [group_1['total_active_mins'], group_2['total_active_mins']],  #
    labels=['Group 1 (Control)', 'Group 2 (Treatment)'],
    showmeans=True
)
plt.title('Box Plot of Total Active Minutes by Group', fontsize=14)
plt.ylabel('Total Active Minutes', fontsize=12)
plt.xlabel('Groups', fontsize=12)
plt.show()
```

- **4. Are there any outliers?**

Yes, when I sort the total_merged_data.csv in descending order, I notice that some values in the tota_active_mins column are as high as 1,121,783 minutes while most numbers are no more than 3000, which is wild. These are outliers.



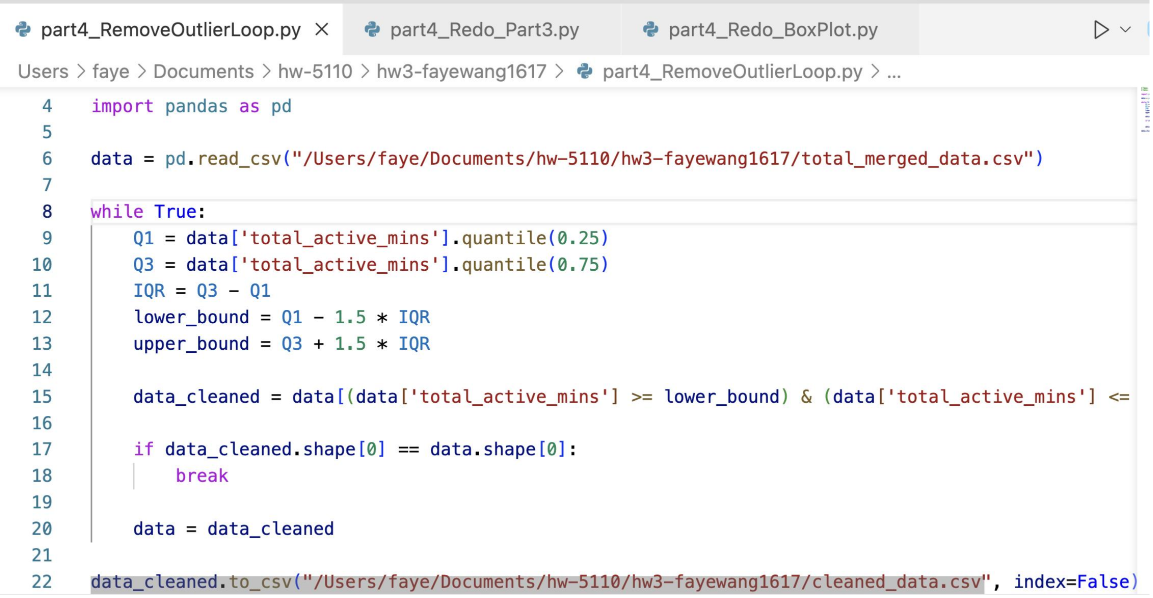
uid	variant_number	total_active_mins
8441	0	1121783
19136	0	937182
21583	0	933492
37805	0	932827
22043	0	833314
12922	0	819211

- **5. What might be causing those outliers? (Hint, look at the data in t1. What is the maximum time a user should possibly have?).**

Those outliers are probably caused by data recording or processing errors, like system glitches, or miscalculated times. Based on the data in t1, the maximum time a user could realistically have in one day is 1,440 minutes (24 hours). Anything way above that is definitely an outlier.

- **6. Remove any data point that might be causing outliers.**

Here is the code:

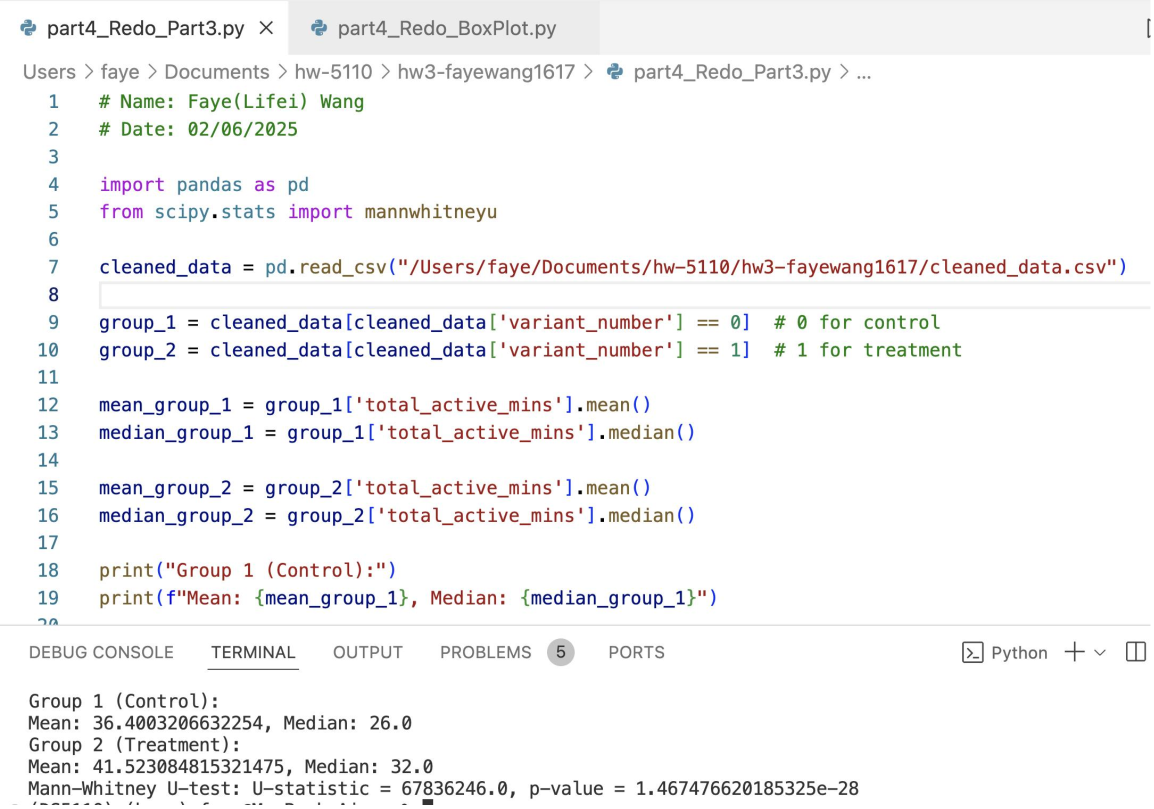

part4_RemoveOutlierLoop.py
part4_Redo_Part3.py
part4_Redo_BoxPlot.py
Users > faye > Documents > hw-5110 > hw3-fayewang1617 > part4_RemoveOutlierLoop.py > ...

```python
import pandas as pd

data = pd.read_csv("/Users/faye/Documents/hw-5110/hw3-fayewang1617/total_merged_data.csv")

while True:
    Q1 = data['total_active_mins'].quantile(0.25)
    Q3 = data['total_active_mins'].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR

    data_cleaned = data[(data['total_active_mins'] >= lower_bound) & (data['total_active_mins'] <=

    if data_cleaned.shape[0] == data.shape[0]:
        break

    data = data_cleaned

data_cleaned.to_csv("/Users/faye/Documents/hw-5110/hw3-fayewang1617/cleaned_data.csv", index=False)
```

And the cleaned data is stored in the cleaned_data.csv now.

- **7. Redo part 2 and 3 with the new data without those data points.**

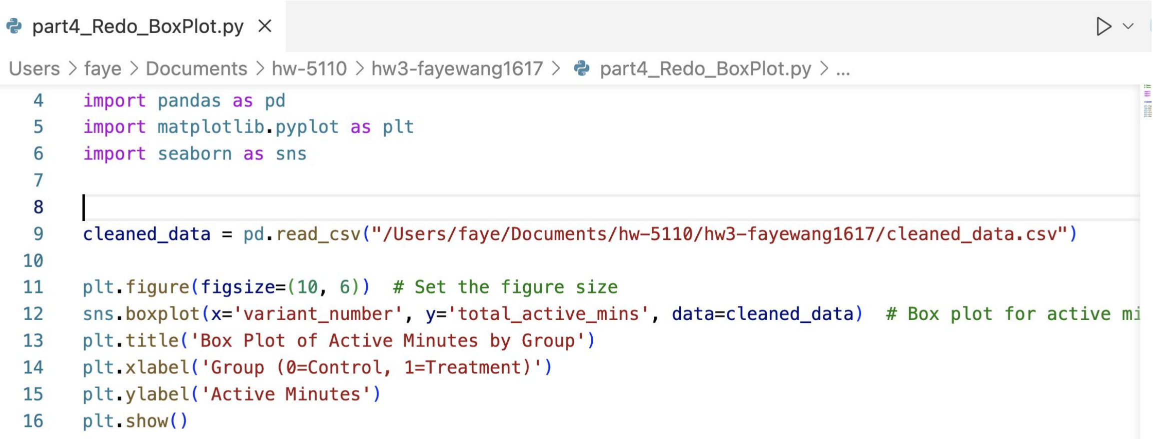
To redo the analysis, I first removed outliers from the dataset using the IQR method. This ensured the cleaned data accurately represents typical user behavior. I then recalculated the mean, median, and ran a Mann-Whitney U test to assess group differences.



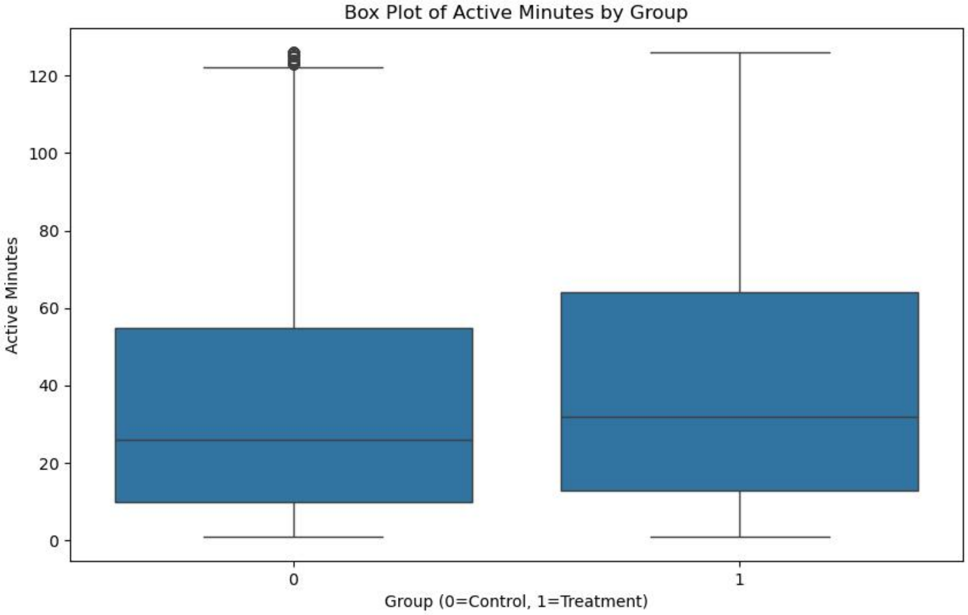
```python
# Name: Faye(Lifei) Wang
# Date: 02/06/2025

import pandas as pd
from scipy.stats import mannwhitneyu

cleaned_data = pd.read_csv("/Users/faye/Documents/hw-5110/hw3-fayewang1617/cleaned_data.csv")

group_1 = cleaned_data[cleaned_data['variant_number'] == 0]  # 0 for control
group_2 = cleaned_data[cleaned_data['variant_number'] == 1]  # 1 for treatment

mean_group_1 = group_1['total_active_mins'].mean()
median_group_1 = group_1['total_active_mins'].median()

mean_group_2 = group_2['total_active_mins'].mean()
median_group_2 = group_2['total_active_mins'].median()

print("Group 1 (Control):")
print(f"Mean: {mean_group_1}, Median: {median_group_1}")
```

```
Group 1 (Control):
Mean: 36.4003206632254, Median: 26.0
Group 2 (Treatment):
Mean: 41.523084815321475, Median: 32.0
Mann-Whitney U-test: U-statistic = 67836246.0, p-value = 1.467476620185325e-28
```

Then I recreated the box plot using the cleaned data to compare active minutes between the control (Group 0) and treatment (Group 1) groups. This updated plot excludes most outliers, making the comparison more representative of the typical user behavior.


part4_Redo_BoxPlot.py
Users > faye > Documents > hw-5110 > hw3-fayewang1617 > part4_Redo_BoxPlot.py > ...

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns


cleaned_data = pd.read_csv("/Users/faye/Documents/hw-5110/hw3-fayewang1617/cleaned_data.csv")

plt.figure(figsize=(10, 6))  # Set the figure size
sns.boxplot(x='variant_number', y='total_active_mins', data=cleaned_data)  # Box plot for active mi
plt.title('Box Plot of Active Minutes by Group')
plt.xlabel('Group (0=Control, 1=Treatment)')
plt.ylabel('Active Minutes')
plt.show()
```



- **8. What is the new conclusion based on the new data?**

After removing outliers, the analysis shows that the treatment group (Group 1) had higher active minutes compared to the control group (Group 0). The mean active minutes for the treatment group increased to 41.52, with a median of 32.0, while the control group had a mean of 36.40 and a median of 26.0.

The Mann-Whitney U test results (U-statistic = 67836246.0, p-value = 1.467e-28) indicate a statistically significant difference between the two groups. This suggests that the treatment had a measurable impact on user activity.

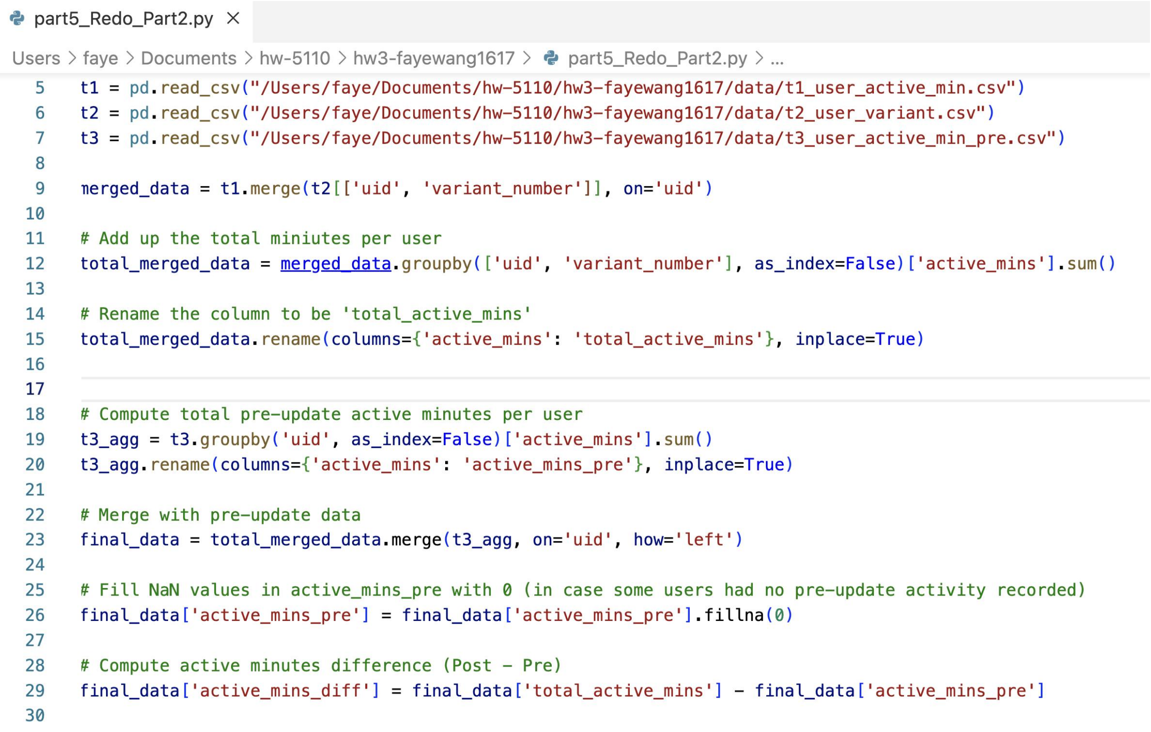
Part 5: Digging Even Deeper (25 Points)

- **1. Why do we care about the data from t3?**

The data from t3_users_active_mins_pre.csv contains the pre-update activity of users on the platform, which is crucial for our analysis. It allows us to establish a baseline for each user's engagement before they were exposed to the new layout. Without this information, our statistical tests may be biased, as differences in user behavior could be due to pre-existing variations rather than the update itself.

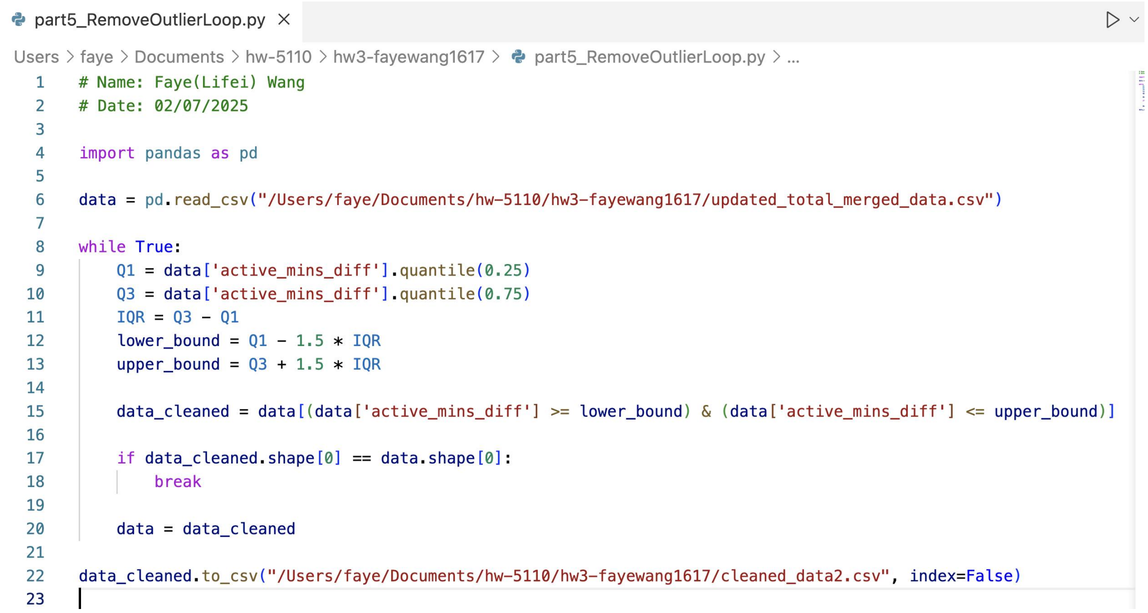
- **2. Accounting for the data from t3 rerun part 2 and 3.**

I merged user activity data, linking post-update (t1), pre-update (t3), and group assignment (t2). It calculates total engagement before and after the update, fills missing pre-update values with 0, and computes the activity change (post – pre) to ensure unbiased analysis.


part5_Redo_Part2.py
Users > faye > Documents > hw-5110 > hw3-fayewang1617 > part5_Redo_Part2.py > ...

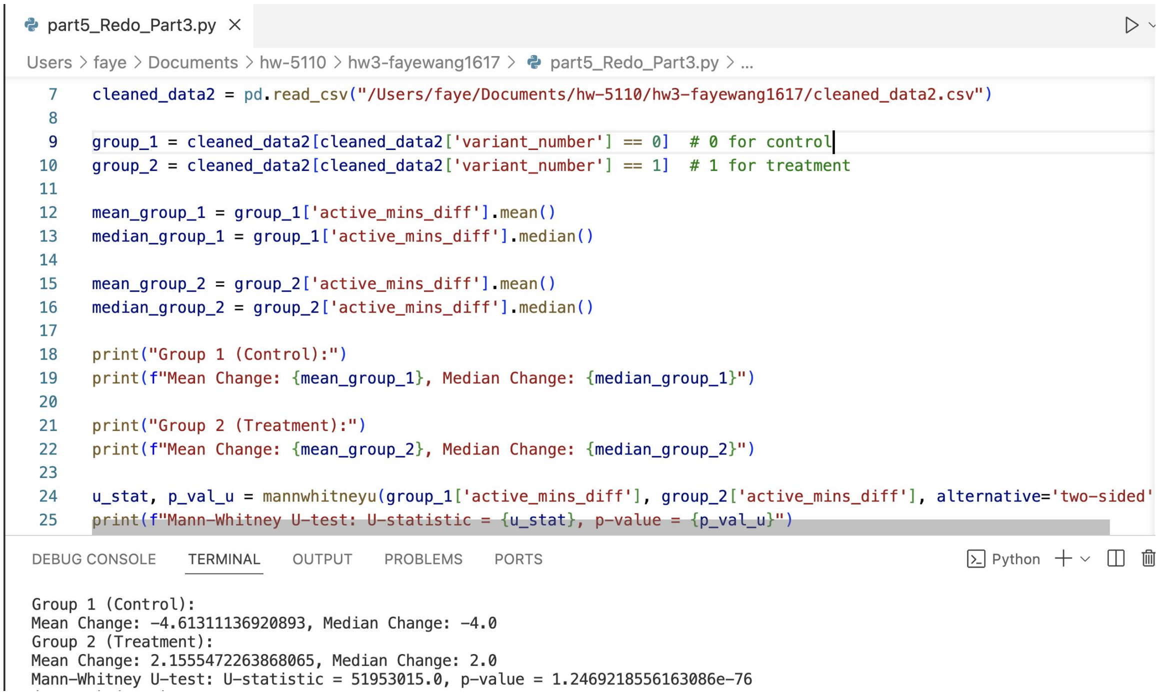
```
t1 = pd.read_csv("/Users/faye/Documents/hw-5110/hw3-fayewang1617/data/t1_user_active_min.csv")
t2 = pd.read_csv("/Users/faye/Documents/hw-5110/hw3-fayewang1617/data/t2_user_variant.csv")
t3 = pd.read_csv("/Users/faye/Documents/hw-5110/hw3-fayewang1617/data/t3_user_active_min_pre.csv")

merged_data = t1.merge(t2[['uid', 'variant_number']], on='uid')

# Add up the total miniutes per user
total_merged_data = merged_data.groupby(['uid', 'variant_number'], as_index=False)['active_mins'].sum()

# Rename the column to be 'total_active_mins'
total_merged_data.rename(columns={'active_mins': 'total_active_mins'}, inplace=True)


# Compute total pre-update active minutes per user
t3_agg = t3.groupby('uid', as_index=False)['active_mins'].sum()
t3_agg.rename(columns={'active_mins': 'active_mins_pre'}, inplace=True)

# Merge with pre-update data
final_data = total_merged_data.merge(t3_agg, on='uid', how='left')

# Fill NaN values in active_mins_pre with 0 (in case some users had no pre-update activity recorded)
final_data['active_mins_pre'] = final_data['active_mins_pre'].fillna(0)

# Compute active minutes difference (Post - Pre)
final_data['active_mins_diff'] = final_data['total_active_mins'] - final_data['active_mins_pre']

```

Removed the outliers again.


part5_RemoveOutlierLoop.py
Users > faye > Documents > hw-5110 > hw3-fayewang1617 > part5_RemoveOutlierLoop.py > ...

```python
# Name: Faye(Lifei) Wang
# Date: 02/07/2025

import pandas as pd

data = pd.read_csv("/Users/faye/Documents/hw-5110/hw3-fayewang1617/updated_total_merged_data.csv")

while True:
    Q1 = data['active_mins_diff'].quantile(0.25)
    Q3 = data['active_mins_diff'].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR

    data_cleaned = data[(data['active_mins_diff'] >= lower_bound) & (data['active_mins_diff'] <= upper_bound)]

    if data_cleaned.shape[0] == data.shape[0]:
        break

    data = data_cleaned

data_cleaned.to_csv("/Users/faye/Documents/hw-5110/hw3-fayewang1617/cleaned_data2.csv", index=False)

```

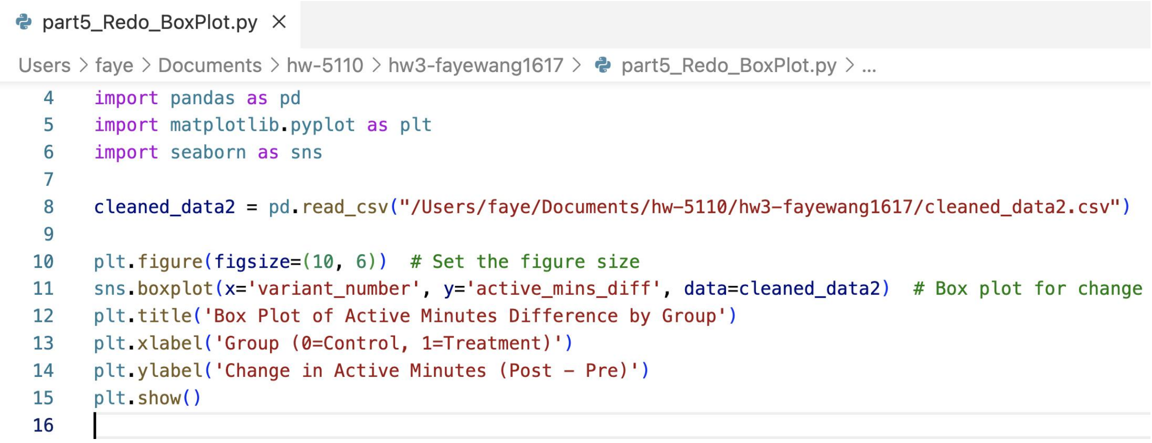
I then recalculated the mean, median, and ran a Mann-Whitney U test to assess group differences.


part5_Redo_Part3.py
Users > faye > Documents > hw-5110 > hw3-fayewang1617 > part5_Redo_Part3.py > ...
DEBUG CONSOLE
TERMINAL
OUTPUT
PROBLEMS
PORTS
Python

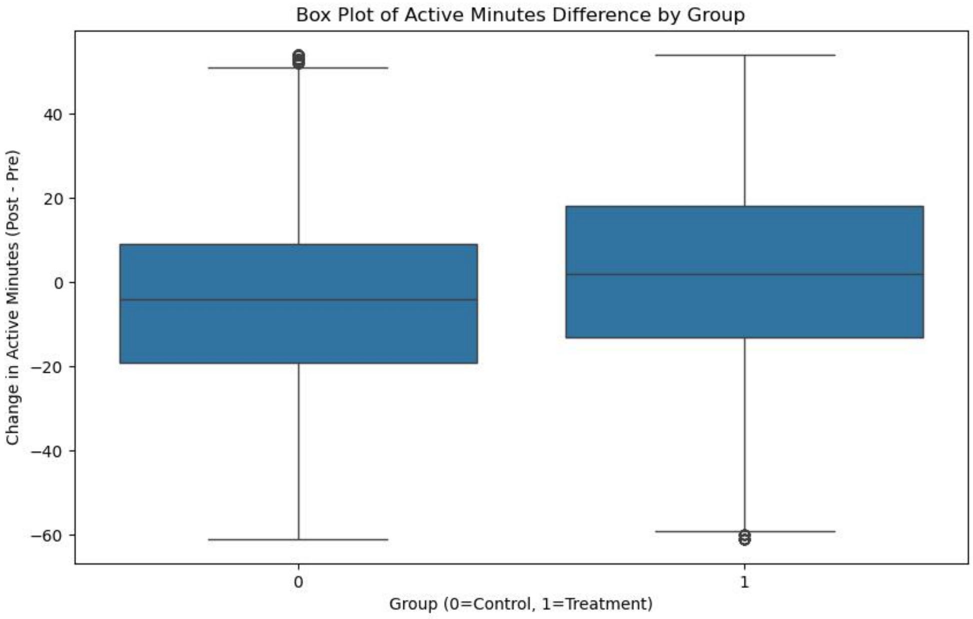
```python
cleaned_data2 = pd.read_csv("/Users/faye/Documents/hw-5110/hw3-fayewang1617/cleaned_data2.csv")

group_1 = cleaned_data2[cleaned_data2['variant_number'] == 0]  # 0 for control
group_2 = cleaned_data2[cleaned_data2['variant_number'] == 1]  # 1 for treatment

mean_group_1 = group_1['active_mins_diff'].mean()
median_group_1 = group_1['active_mins_diff'].median()

mean_group_2 = group_2['active_mins_diff'].mean()
median_group_2 = group_2['active_mins_diff'].median()

print("Group 1 (Control):")
print(f"Mean Change: {mean_group_1}, Median Change: {median_group_1}")

print("Group 2 (Treatment):")
print(f"Mean Change: {mean_group_2}, Median Change: {median_group_2}")

u_stat, p_val_u = mannwhitneyu(group_1['active_mins_diff'], group_2['active_mins_diff'], alternative='two-sided'
print(f"Mann-Whitney U-test: U-statistic = {u_stat}, p-value = {p_val_u}")
```

```
Group 1 (Control):
Mean Change: -4.61311136920893, Median Change: -4.0
Group 2 (Treatment):
Mean Change: 2.1555472263868065, Median Change: 2.0
Mann-Whitney U-test: U-statistic = 51953015.0, p-value = 1.2469218556163086e-76
```

Then I recreated the box plot again using the cleaned data to compare active_mins_diff between the control (Group 0) and treatment (Group 1) groups.


part5_Redo_BoxPlot.py ×
Users > faye > Documents > hw-5110 > hw3-fayewang1617 > part5_Redo_BoxPlot.py > ...

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

cleaned_data2 = pd.read_csv("/Users/faye/Documents/hw-5110/hw3-fayewang1617/cleaned_data2.csv")

plt.figure(figsize=(10, 6))  # Set the figure size
sns.boxplot(x='variant_number', y='active_mins_diff', data=cleaned_data2)  # Box plot for change
plt.title('Box Plot of Active Minutes Difference by Group')
plt.xlabel('Group (0=Control, 1=Treatment)')
plt.ylabel('Change in Active Minutes (Post - Pre)')
plt.show()

```



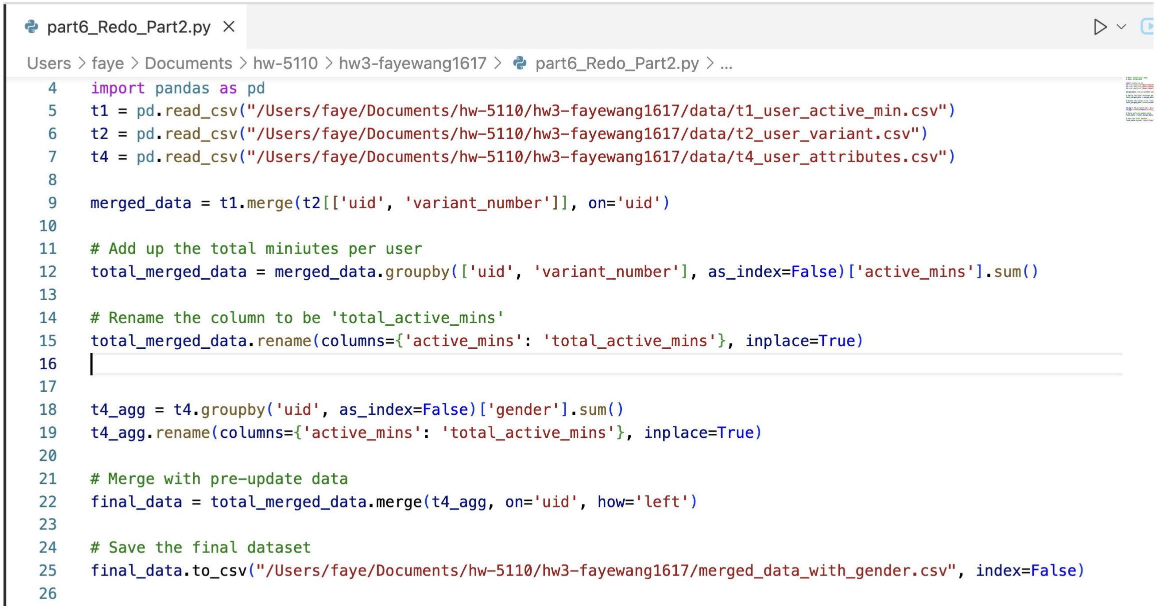
- **3. Are there any new conclusion?**

With the new data from t3, we see that users with the new layout spent more time (+2.16 min on average, +2.0 min median), while users with the old layout spent less (−4.61 min on average, −4.0 min median).

The test result (p = 1.25e−76) shows this difference is significant, not just random. The boxplot also shows the new layout led to more time spent.

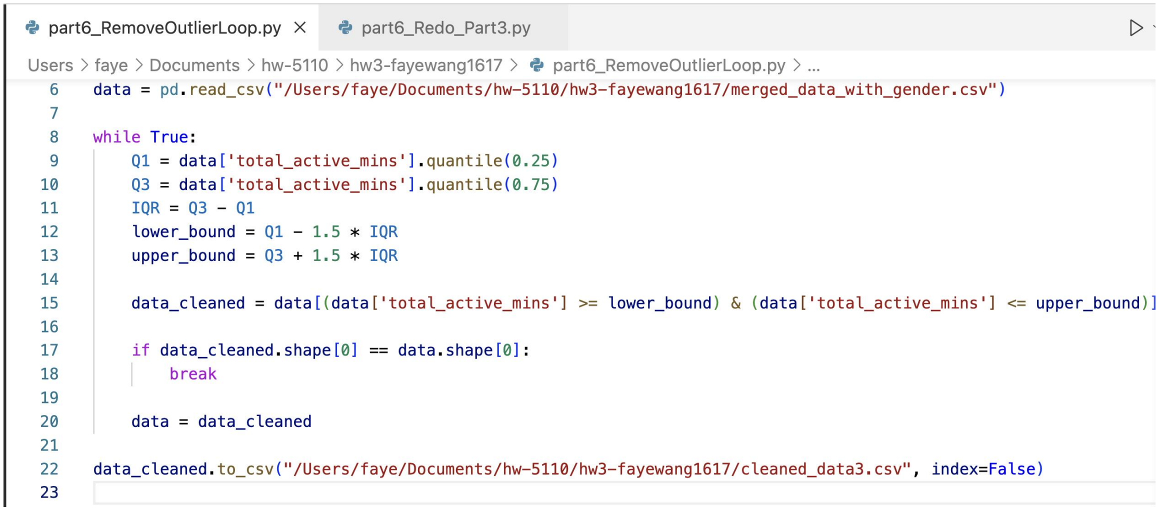
Part 6: Exploring other conclusions (10 Points)

I merged user activity data, linking t1, t2, and t4. It analyzed the impact of gender on user engagement by segmenting the data into male and female groups.


part6_Redo_Part2.py
Users > faye > Documents > hw-5110 > hw3-fayewang1617 > part6_Redo_Part2.py > ...

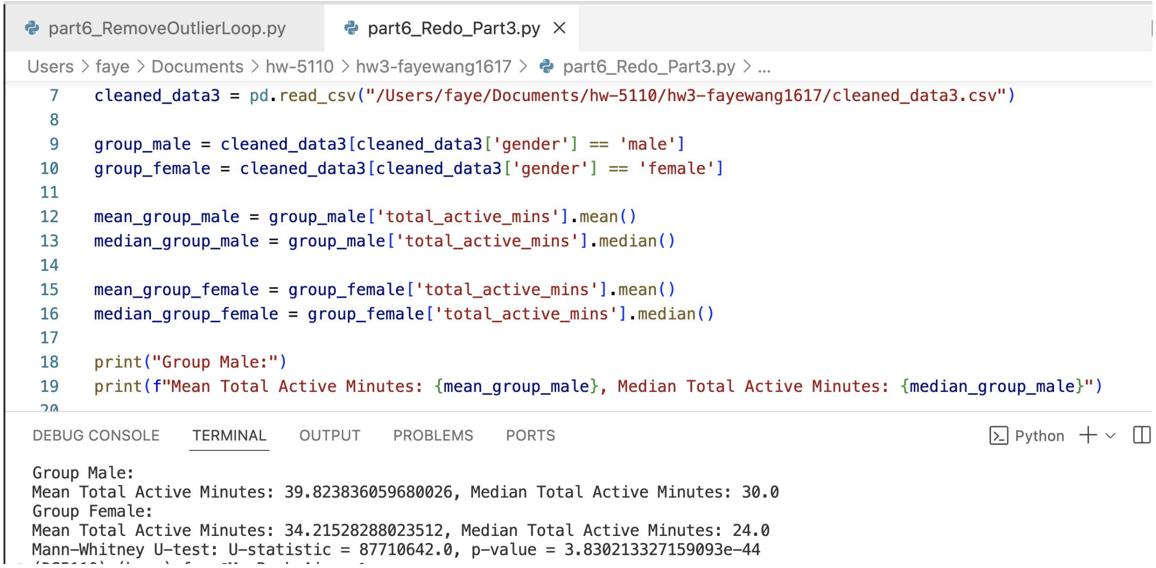
```python
import pandas as pd
t1 = pd.read_csv("/Users/faye/Documents/hw-5110/hw3-fayewang1617/data/t1_user_active_min.csv")
t2 = pd.read_csv("/Users/faye/Documents/hw-5110/hw3-fayewang1617/data/t2_user_variant.csv")
t4 = pd.read_csv("/Users/faye/Documents/hw-5110/hw3-fayewang1617/data/t4_user_attributes.csv")

merged_data = t1.merge(t2[['uid', 'variant_number']], on='uid')

# Add up the total miniutes per user
total_merged_data = merged_data.groupby(['uid', 'variant_number'], as_index=False)['active_mins'].sum()

# Rename the column to be 'total_active_mins'
total_merged_data.rename(columns={'active_mins': 'total_active_mins'}, inplace=True)


t4_agg = t4.groupby('uid', as_index=False)['gender'].sum()
t4_agg.rename(columns={'active_mins': 'total_active_mins'}, inplace=True)

# Merge with pre-update data
final_data = total_merged_data.merge(t4_agg, on='uid', how='left')

# Save the final dataset
final_data.to_csv("/Users/faye/Documents/hw-5110/hw3-fayewang1617/merged_data_with_gender.csv", index=False)

```

Removed the outliers again.


part6_RemoveOutlierLoop.py
part6_Redo_Part3.py
Users > faye > Documents > hw-5110 > hw3-fayewang1617 > part6_RemoveOutlierLoop.py > ...

```python
data = pd.read_csv("/Users/faye/Documents/hw-5110/hw3-fayewang1617/merged_data_with_gender.csv")

while True:
    Q1 = data['total_active_mins'].quantile(0.25)
    Q3 = data['total_active_mins'].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR

    data_cleaned = data[(data['total_active_mins'] >= lower_bound) & (data['total_active_mins'] <= upper_bound)]

    if data_cleaned.shape[0] == data.shape[0]:
        break

    data = data_cleaned

data_cleaned.to_csv("/Users/faye/Documents/hw-5110/hw3-fayewang1617/cleaned_data3.csv", index=False)
```

I then recalculated the mean, median, and ran a Mann-Whitney U test to assess group differences.


part6_RemoveOutlierLoop.py
part6_Redo_Part3.py
Users > faye > Documents > hw-5110 > hw3-fayewang1617 > part6_Redo_Part3.py > ...
DEBUG CONSOLE
TERMINAL
OUTPUT
PROBLEMS
PORTS
Python

```python
cleaned_data3 = pd.read_csv("/Users/faye/Documents/hw-5110/hw3-fayewang1617/cleaned_data3.csv")

group_male = cleaned_data3[cleaned_data3['gender'] == 'male']
group_female = cleaned_data3[cleaned_data3['gender'] == 'female']

mean_group_male = group_male['total_active_mins'].mean()
median_group_male = group_male['total_active_mins'].median()

mean_group_female = group_female['total_active_mins'].mean()
median_group_female = group_female['total_active_mins'].median()

print("Group Male:")
print(f"Mean Total Active Minutes: {mean_group_male}, Median Total Active Minutes: {median_group_male}")
```

```
Group Male:
Mean Total Active Minutes: 39.823836059680026, Median Total Active Minutes: 30.0
Group Female:
Mean Total Active Minutes: 34.21528288023512, Median Total Active Minutes: 24.0
Mann-Whitney U-test: U-statistic = 87710642.0, p-value = 3.830213327159093e-44
```

The data shows that male users spent more time on the platform (39.82 min on average, 30.0 min median), while female users spent less (34.22 min on average, 24.0 min median).

Also, the test result (p = 3.83e−44) shows this difference is highly significant, meaning gender plays a role in engagement levels. The data suggests that male users were generally more active than female users.

Part 7: Summarize Your Results(10 Points)

- **Part 1:**

We first explored the structure of the datasets, including post-update activity (t1), experiment groups (t2), pre-update activity (t3), and user attributes (t4). This step was essential to understand the available data and ensure proper merging. Early observations revealed skewed distributions in active minutes, hinting at potential outliers.

- **Part 2:**

To analyze engagement differences between control (variant_number = 0) and treatment (variant_number = 1), we merged t1 with t2, linking users by uid. This created a unified dataset of "total post-update active minutes per user". Without correctly grouping users, we wouldn't be able to isolate the effects of the new design.

- **Part 3:**

The data showed distinct differences in mean and median values between the two groups (Control group: 837.64 vs 52.0 minutes; Treatment group: 784.20 vs 71.0 minutes), reflecting a highly skewed distribution. Although the U-test results (U-statistic = 158,271,912.0; p-value = 7.64e−34) confirmed a statistically significant difference between the two groups, highly skewed distribution suggested that outliers might be distorting the results, requiring further investigation.

- **Part 4:**

Boxplots confirmed what I suspected—some users had 1,121,783 minutes logged in while most numbers are no more than 3000, which is wild. I removed these outliers using the IQR method, and suddenly, the results changed. After cleaning, the treatment group showed a higher mean (41.5 vs. 36.4) and median (32 vs. 26) than the control, making it clear that the new layout increased engagement. This showed how much outliers can distort conclusions if not handled properly.

- **Part 5:**

At this point, I realized that just looking at post-update activity wasn't enough—I needed to know how much users were already using the platform before the redesign. After merging t3, I recalculated everything based on engagement change (post – pre). Now, the treatment group had a +2.16 min/day increase, while the control group lost –4.61 min/day. The U-test (p ≈ 0) confirmed that the new layout not only worked, but also prevented a decline in usage.

- **Part 6:**

Finally, I wanted to see if gender affected engagement. After merging t4 (user attributes), I found that male users spent significantly more time on the platform than female users (p ≈ 0). This suggests that the redesign's impact wasn't the same for everyone, which could be important for future product decisions.

- **Conclusion**

This assignment showed me how messy real-world data can be and how important it is to clean and structure it properly before making conclusions. Initially, the data suggested that the new design might not be effective, but after removing outliers and considering pre-update behavior, the results flipped, showing a real impact. Breaking the data down further (like by gender) also proved useful, as engagement wasn't equal across all users.

In the end, data analysis isn’t just about running tests—it’s about making sure you’re asking the right questions, cleaning your data properly, and looking at it from different angles to get the full picture.