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

1. What data is in file "t1_users_active_mins.csv"?

This file tracks individual user activity in minutes across various dates **after the experiment started**. Each row represents the total number of minutes a user spent on the platform on a specific date. If a user never visited the platform on a certain date, there will be no data for that user on that date.

- **Columns**:
    - uid: The unique identifier for users.
    - dt: The date when the corresponding active minutes are registered (stored as strings).
    - active_mins: The number of minutes the user was active on the platform for a given date.
- **Shape**:
    - The dataset contains **1,066,403 rows** and **3 columns**.
- **Data Types**:
    - uid: Integer (int64).
    - dt: Object (string), representing date values.
    - active_mins: Float (float64), representing the number of active minutes.

2. What data is in file "t2_users_variant.csv"?

This file contains users' treatment assignment information, identifying whether they belong to the control or treatment group. Each row corresponds to a unique user and includes their experiment group assignment, the date they entered the experiment, and their signup date.

- **Columns**:
    - uid: The unique identifier for users.
    - variant_number: A numeric code indicating the group the user belongs to:
        - 0 for the control group (did not receive the new update).
        - 1 for the treatment group (received the new update).
    - dt: The date when the user entered the experiment (all entries should have the date 2019-02-06).
    - signup_date: The date when the user signed up for the platform (stored as strings).
- **Shape**:

    - The dataset contains **50,001 rows** and **4 columns**.
- **Data Types**:
    - uid: Integer (int64), representing user IDs.
    - variant_number: Integer (int64), representing the experiment group.
    - dt: Object (string), representing the date of variant assignment.
    - signup_date: Object (string), representing the user's signup date.

3. What data is in file "t3_users_active_mins_pre.csv"?

This file tracks individual user activity in minutes across various dates **before the experiment started**. It is similar in format to t1_users_active_mins.csv, but the date range extends to the period before the experiment began.

- **Columns**:
    - uid: The unique identifier for users.
    - dt: The date when the corresponding active minutes are registered (stored as strings).
    - active_mins: The number of minutes the user was active on the platform for a given date.
- **Shape**:
    - The dataset contains **1,190,094 rows** and **3 columns**.
- **Data Types**:
    - uid: Integer (int64), representing user IDs.
    - dt: Object (string), representing the date of activity.
    - active_mins: Float (float64), representing the number of active minutes.

4. What data is in file "t4_users_attributes.csv"?

This file contains demographic information and user categorization. Each row represents attributes of a unique user. It is valuable for segmenting users and analyzing behavior based on gender and activity level.

- **Columns**:
    - uid: The unique identifier for users.
    - gender: The gender of the user, represented as strings (male, female, or unknown).
    - user_type: The type of user, represented as strings. Possible values are:
        - new_user

        - non_reader
        - reader
        - contributor

- **Shape**:

    - The dataset contains **50,001 rows** and **3 columns**.

- **Data Types**:

    - uid: Integer (int64), representing user IDs.
    - gender: Object (string), representing the user's gender.
    - user_type: Object (string), representing the activity-level segment of the user.

5. What data is in file "table_schema.txt"?

t1_user_active_min.csv:

This table logs active minutes after the experiment started.

Each row represents the total minutes spent on the site by a user on a specific date.

If a user did not visit the site on a certain date, no data is recorded for that date.

Columns:

uid: User ID.

dt: Date of activity (active minutes registered).

active_mins: Total minutes spent on the site for that date.

t2_user_variant.csv:

This table contains users' treatment assignment information.

Each row corresponds to a unique user and their experiment group.

Columns:

uid: User ID.

variant_number: Experiment variant (0 = control, 1 = treatment).

dt: Date when the user entered the experiment (all users' date = '2019-02-06').

signup_date: The date the user signed up for the platform.

t3_user_active_min_pre.csv:

This table logs active minutes before the experiment started.

The format is similar to t1_user_active_min.csv, except the date range extends before the experiment start date.

Columns:

uid: User ID.

dt: Date of activity (active minutes registered).

active_mins: Total minutes spent on the site for that date.

t4_user_attributes.csv:

This table contains attributes for users.

Each row represents the attributes of a unique user.

Columns:

uid: User ID.

user_type: Activity-level segment (can be new_user, non_reader, reader, or contributor).

gender: Gender of the user (can be male, female, or unknown).

Part 2: Organizing the Data (15 Points)

1.What is the overall objective of this study?

The objective of this study is to determine whether the new layout and features of the social media platform increase user engagement by measuring the total time users spend on the website. If the update proves effective, the company will invest in advertisements and fully launch the new version. To achieve this, we will compare user activity between the control group (old version) and the treatment group (new version) and conduct statistical analysis to assess whether the difference in playtime is statistically significant.

2.What data do we need to reach the objective?

The objective is to determine whether the new platform update increases the total time users spend on the website. To analyze this, we need:

1. **User activity data**: The total minutes each user spent on the platform.
2. **Experiment group assignment**: Whether a user is in the control group (variant_number = 0) or the treatment group (variant_number = 1).

3. **Time segmentation**: Ideally, we would compare **before** and **after** the update. However, based on the schema, t1_user_active_min.csv only contains post-experiment activity, meaning we can only analyze the **total time spent after the update**.

Thus, the required data consists of:

- **uid** (user ID)
- **variant_number** (experimental group: control vs. treatment)
- **total_act_mins** (total time spent after the update)

3.How is the data in t1_user_active_min.csv organized?

Each row represents a **single user's activity on a specific date**.

Columns:

- uid: Unique user ID.
- dt: The date when the activity was recorded.
- active_mins: The number of minutes the user spent on the platform on that date.

Key characteristics:

- A user may have multiple entries (one per day).
- If a user **did not visit the platform** on a particular date, there is **no entry** for that day.

4.How should the data be organized to be useful?

To analyze the effect of the update, we need to **aggregate user activity over time**:

1. **Summarize activity per user**:
    - Since we only have post-experiment data, we should **sum** active_mins for each uid to get their **total activity minutes after the update**.
    - This removes the dt column since the total time matters more than daily records.
2. **Merge with t2_user_variant.csv**:
    - We need to associate each user with their variant_number to know whether they were in the **control group or treatment group**.

5.Organize it.

- Create a Python file (2.py) to process t1_user_active_min.csv and t2_user_variant.csv.
- Aggregate user activity by summing the total minutes each user spent on the platform after the update.

- Merge the aggregated data with t2_user_variant.csv to match users with their experiment group (variant_number).
- Store the final organized data in a new CSV file (organized_user_data.csv) inside the part2 folder.
- ```
  part2 > organized_user_data.csv
  uid,total_act_mins,variant_number
  0,43.0,0
  1,15205.0,0
  2,17.0,0
  3,77.0,0
  4,39.0,0
  5,174.0,0
  6,26.0,0
  7,21.0,0
  9,42.0,0
  10,127.0,0
  11,142.0,0
  13,17.0,0
  14,3.0,0
  15,35.0,0
  16,56.0,0
  17,5462.0,0
  18,2.0,0
  19,551.0,0
  20,77.0,0
  21,11.0,0
  22,19.0,0
  23,58.0,0
  24,130.0,0
  25,359.0,0
  ```

Part 3: Statistical Analysis (10 points)

1.Is there a statistical difference between group 1 and group 2?

```
Statistical Analysis Results:
========================================
Scipy T-Test Result: TtestResult(statistic=np.float64(0.32346507126292273), pvalue=np.float64(0.7463445065262613), df=np.flo
at64(46631.0))
========================================
Pingouin T-Test Result:
                T           dof alternative     p-val              CI95%   cohen-d   BF10    power
T-test  0.405609  20207.387288   two-sided  0.685034  [-204.81, 311.69]  0.003763  0.014  0.06207
========================================
Statsmodels T-Test Result: (np.float64(0.32346507126292273), np.float64(0.7463445065262613), np.float64(46631.0))
========================================
Manual T-Test Result (NumPy): t-statistic = 0.4056
========================================
ResearchPy T-Test Result:
                                 Independent t-test     results
0  Difference (total_act_mins - total_act_mins) =      53.4400
1                             Degrees of freedom =   46631.0000
2                                              t =       0.3235
3                          Two side test p value =       0.7463
4                         Difference < 0 p value =       0.6268
5                         Difference > 0 p value =       0.3732
6                                      Cohen's d =       0.0038
7                                      Hedge's g =       0.0038
8                                  Glass's delta1 =       0.0036
9                                Point-Biserial r =       0.0015
```

- To determine whether there is a statistically significant difference between the control group and the treatment group, I applied the t-test method using five different approaches (Scipy, Pingouin, Statsmodels, NumPy manual calculation, and ResearchPy).
- The t-test compares the means of the two groups to check if the difference is statistically significant. The results are stored in 3.py in the part3 folder.
```

According to the results from the five methods, the p-values 0.685034 (by the Pingouin) and also 0.7463445065262613 (by the Statsmodels) , which are both greater than 0.05 (the other methods' p-values are also greater that 0.05).

Since p-value > 0.05, I think we fail to reject the null hypothesis, meaning there is no statistically significant difference between the control and treatment groups.

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

```
========================================
Control Group - Mean: 837.64, Median: 52.00
Treatment Group - Mean: 784.20, Median: 71.00
```

See the 3.py file in the part3 folder for the calculations and I also post the screenshot for the output.

Group 1: Control Group

- Mean: 837.64
- Median: 52.00

Group 2: Treatment Group

- Mean: 784.20
- Median: 71.00

3.What can you conclude based on that data?

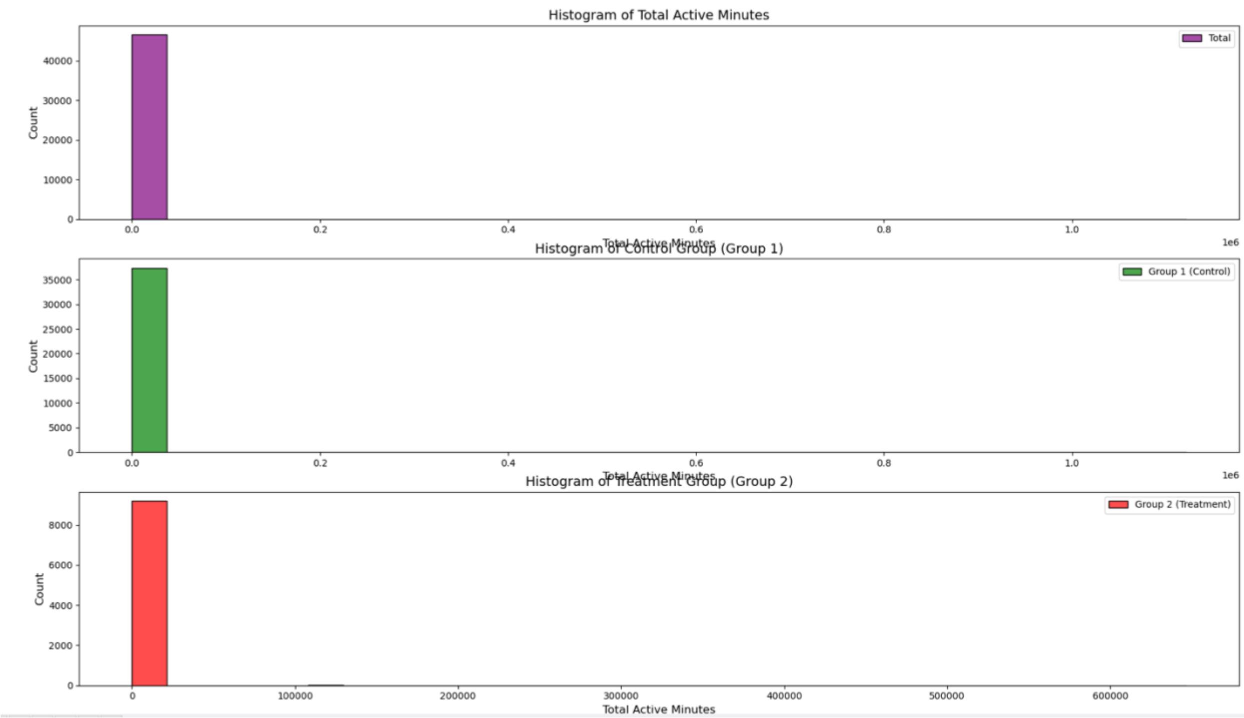
- Median Comparison: The control group has a lower median (52.00) compared to the treatment group (71.00). However, considering that there are 40,000 users in the control group and 10,000 users in the treatment group, this median difference does not necessarily imply a significant effect of the update.
- Mean Comparison: The mean values of Group 1 (837.64) and Group 2 (784.20) are relatively close, meaning that the total playing time in both groups is similar. This suggests that the update does not have a significant effect on user engagement.
- Based on the statistical analysis, I think we cannot conclude that the new version of the platform increases total playing time compared to the old version. The update does not appear to have had a measurable impact on user engagement.

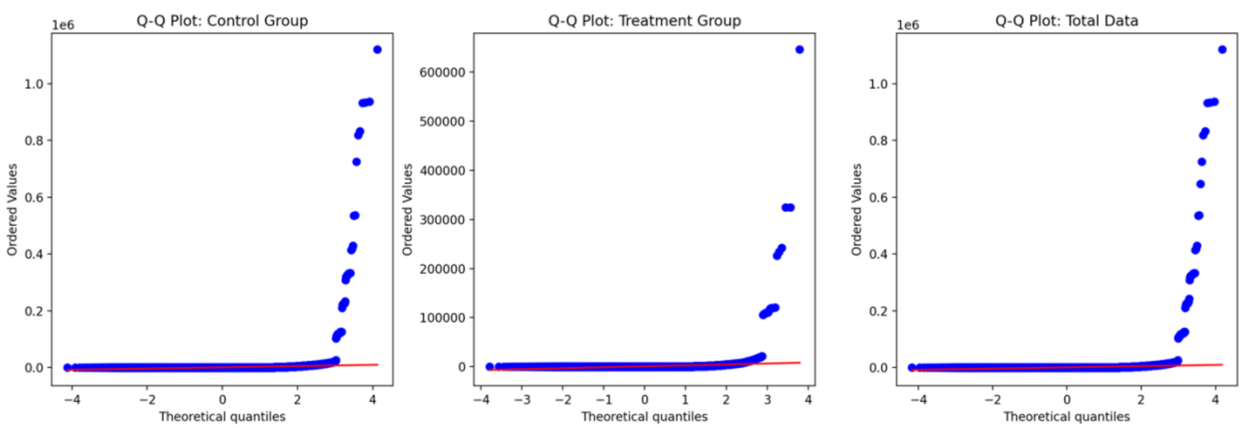
Part 4: Digging a Little Deeper (25 Points)

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

No, I think we cannot fully trust the results without deeper validation. The unequal sample sizes (40,000 vs. 10,000) may influence statistical outcomes, and high variability in user engagement suggests potential outliers affecting the mean. Additionally, the t-test assumes normality, but user activity data is often skewed, meaning a non-parametric test might be more appropriate. External factors like seasonality or marketing campaigns could also impact user behavior, and total playing time alone may not fully capture engagement changes. To validate our findings, we should explore data distribution, consider alternative metrics, account for external influences, and perform additional statistical tests.

2.Is the data normally distributed?

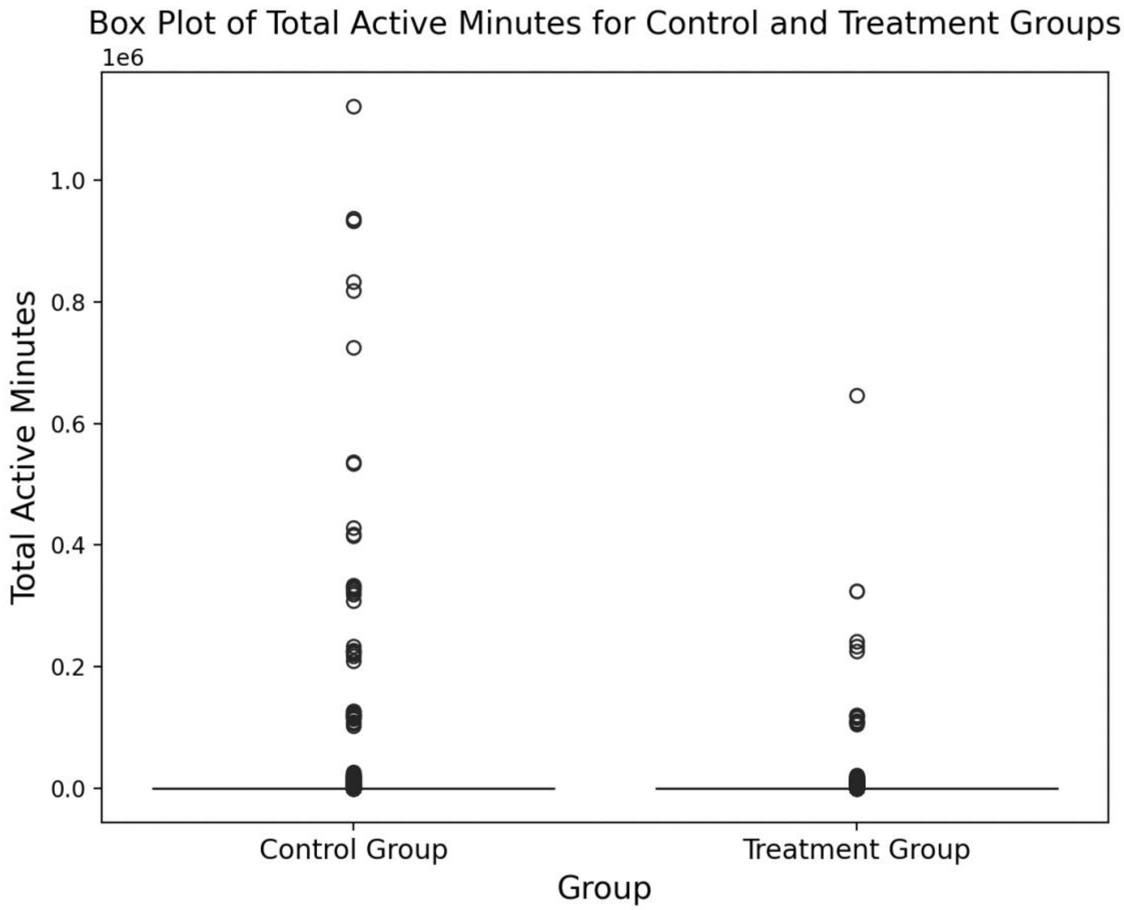




No, the data is not normally distributed. The histograms reveal a highly skewed distribution where most users have low total active minutes, but a few have extremely high values, indicating a long right tail. The Q-Q plots further confirm this, as the data points deviate significantly from the red reference line, particularly in the upper quantiles, where extreme values stretch far beyond what would be expected in a normal distribution. This pattern suggests the presence of heavy-tailed distributions with significant outliers, reinforcing that the data does not conform to normality.

The code was stored in the 4.py in the part4 folder.

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



The code was stored in the 5.py in the part4 folder.

4.Are there any outliers?

Yes, there are significant outliers in both the control and treatment groups, as shown by the numerous black dots above the whiskers in the box plot. These outliers indicate a small subset of users with exceptionally high total active minutes, while most users have relatively low playtime, resulting in a highly skewed distribution. The compact interquartile range suggests that the majority of users exhibit similar behavior, but the extreme outliers may inflate the mean and affect statistical tests like the t-test.

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

```
● (DS5110) (base) ➜  part4 python 6.py
 Highest value in table 1: 99999.0 minutes
 Number of entries exceeding 1440 minutes per day: 172
 Outlier data saved to ../part2/outliers_analysis.csv
 Sample outlier records:
         uid          dt  active_mins
 26670  1219  2019-02-22      99999.0
 38577  1738  2019-02-25      99999.0
 53137  2398  2019-04-19      99999.0
 69731  3124  2019-04-26      99999.0
 69742  3124  2019-05-10      99999.0
```

The outliers are likely caused by data recording errors or anomalies in the logging system. From the dataset, the highest recorded active minutes is 99,999, which is not realistic, as the maximum possible time a user can be active in a day is 24 hours (1,440 minutes). Additionally, 172 entries exceed the 1440-minute (24*60) threshold, further confirming abnormal values. These extreme outliers could be due to bugs in the tracking system, duplicate logs, or erroneous data entries where session durations were incorrectly stored. To ensure data integrity, these anomalies should be investigated, removed, or corrected before conducting further analysis.

The code was stored in the 6.py in the part4 folder.

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

I removed rows in t1_user_active_min.csv where active_mins exceeded 12 hours (720 minutes) per day and stored the cleaned data in the part4 folder as cleaned_data.csv. The reason for choosing 12 hours instead of 24 hours is that, while some highly engaged users may spend extended time on the platform, it is still unrealistic for most users to be active for an entire day. I think a 12-hour threshold balances filtering out erroneous data while keeping legitimate high-engagement users.

I also try 16 hours, but the result is not good, so finally I choose the 12 hours.

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

The reorganized data was stored in the part4 folder (t1_t2_cleaned_data.csv) by running the 7.py.

The code (redo part 2 and part 3) was stored in the 8.py in the part4 folder.

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

```
Statistical Analysis Results:
==========================================
Scipy T-Test Result: TtestResult(statistic=np.float64(0.18252326866141094), pvalue=np.float64(0.855172888760245), df=np.floa
t64(46631.0))
==========================================
Pingouin T-Test Result:
               T            dof alternative     p-val            CI95%   cohen-d   BF10     power
T-test  0.183991  14223.420555   two-sided  0.854023  [-32.59, 39.34]  0.002123  0.013  0.053825
==========================================
Statsmodels T-Test Result: (np.float64(0.1825232686614109), np.float64(0.855172888760245), np.float64(46631.0))
==========================================
Manual T-Test Result (NumPy): t-statistic = 0.1840
==========================================
ResearchPy T-Test Result:
                               Independent t-test     results
0  Difference (total_act_mins - total_act_mins) =      3.3760
1                           Degrees of freedom =   46631.0000
2                                            t =      0.1825
3                        Two side test p value =      0.8552
4                       Difference < 0 p value =      0.5724
5                       Difference > 0 p value =      0.4276
6                                  Cohen's d =      0.0021
7                                  Hedge's g =      0.0021
8                              Glass's delta1 =      0.0021
9                            Point-Biserial r =      0.0008
==========================================
Control Group - Mean: 454.06, Median: 52.00
Treatment Group - Mean: 450.68, Median: 71.00
```

After removing outliers, the new analysis confirms that there is **no** statistically significant difference between the control and treatment groups, as indicated by the high p-value (0.855) and a t-statistic close to zero (0.1825). The effect size (Cohen's d = 0.0021) is extremely small, meaning any observed difference is practically negligible. Additionally, the mean active minutes are nearly identical (Control: 454.06, Treatment: 450.68), further supporting the conclusion that the update had no measurable impact on user engagement. While the median values differ slightly (Control: 52.00, Treatment: 71.00), the statistical tests show that this is likely due to natural variability rather than the platform update. Ultimately, the new features did not lead to a significant increase in total active minutes on the platform.

Part 5: Digging Even Deeper (25 Points)

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

t3_user_active_min_pre.csv is essential because it provides pre-experiment user activity data, allowing us to establish a baseline and compare user behavior before and after the update. Without t3, we can only analyze post-experiment activity, making it difficult to determine if any observed changes were actually caused by the update or just natural variations. By incorporating t3, we can conduct a difference-in-differences analysis, which is more statistically reliable than a simple post-experiment comparison. This helps us detect whether user engagement increased or decreased due to the update, rather than relying on assumptions. Additionally, t3 helps reduce bias by controlling for pre-existing differences in user behavior—without it, we might incorrectly assume that both groups started at the same activity level. Furthermore, it allows us to segment new vs. existing users, helping us understand whether highly engaged users remain active and if less active users increase their engagement. Ultimately, t3 is critical for making accurate conclusions about the platform update's effectiveness, ensuring that our results are not misleading due to external factors or inherent user differences.

2.Accounting for the data from t3 return part 2 and 3.

The reorganized data (t1_t2_t3.csv) was stored in the part5 folder by running 9.py.

The code (accounting for the data from t3 return part 2 and 3) was stored in the 10.py in the part5 folder.

part5 > t1_t2_t3.csv

```
uid,total_act_mins_post,variant_number,total_act_mins_pre
0,43.0,0,70.0
1,15205.0,0,18313.0
2,17.0,0,37.0
3,77.0,0,108.0
4,39.0,0,66.0
5,174.0,0,21.0
6,26.0,0,12.0
7,21.0,0,34.0
9,42.0,0,52.0
10,127.0,0,74.0
11,142.0,0,96.0
13,17.0,0,29.0
14,3.0,0,15.0
15,35.0,0,1523.0
16,56.0,0,9.0
17,5462.0,0,717.0
18,2.0,0,33.0
19,551.0,0,228.0
20,77.0,0,127.0
21,11.0,0,25.0
22,19.0,0,10.0
23,58.0,0,61.0
24,130.0,0,31.0
25,359.0,0,205.0
```

3.Are their any new conclusion?

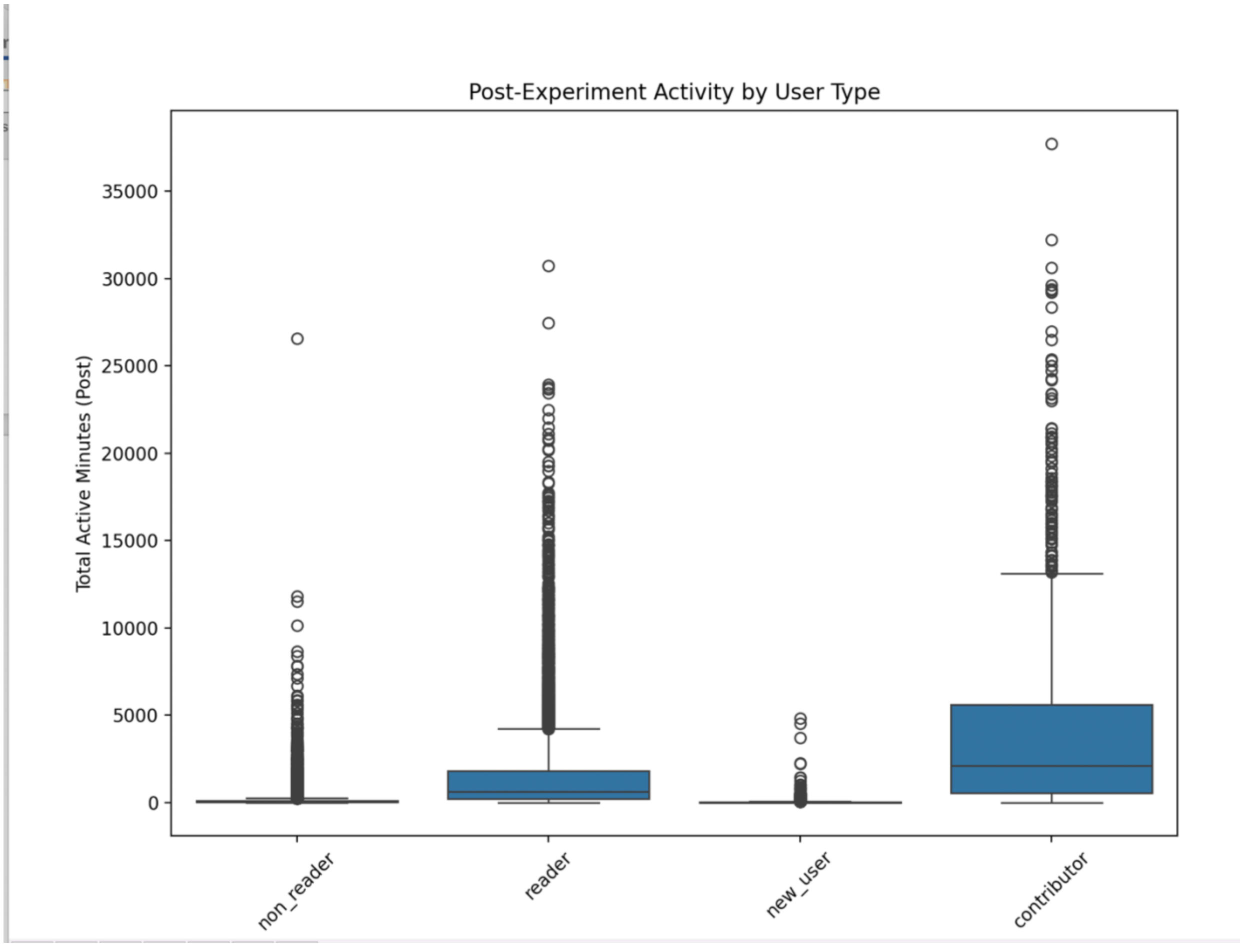
```
Statistical Analysis Results:
========================================
Control Group - Scipy T-Test Result: TtestResult(statistic=np.float64(-9.960705145101219), pvalue=np.float64(2.3545960108079
71e-23), df=np.float64(64282.0))
========================================
Control Group - Pingouin T-Test Result:
                T          dof alternative         p-val           CI95%   cohen-d       BF10 power
T-test -9.960705  64088.69886   two-sided  2.354873e-23  [-11.69, -7.85]  0.078572  2.967e+19   NaN
========================================
Control Group - Statsmodels T-Test Result: (np.float64(-9.960705145101219), np.float64(2.354596010807971e-23), np.float64(64
282.0))
========================================
Control Group - Manual T-Test (NumPy): t-statistic = -9.9607
========================================
Control Group - ResearchPy T-Test Result:
                                   Independent t-test     results
0  Difference (total_act_mins_post - total_act_mi...     -9.7705
1                             Degrees of freedom =   64282.0000
2                                              t =      -9.9607
3                          Two side test p value =       0.0000
4                         Difference < 0 p value =       0.0000
5                         Difference > 0 p value =       1.0000
6                                      Cohen's d =      -0.0786
7                                      Hedge's g =      -0.0786
8                                  Glass's delta1 =      -0.0808
9                                Point-Biserial r =      -0.0393
========================================
Control Group - Mean Pre: 96.53, Post: 86.75
Control Group - Median Pre: 46.00, Post: 39.00
========================================
Treatment Group - Scipy T-Test Result: TtestResult(statistic=np.float64(14.085969883667726), pvalue=np.float64(8.53067433170
4647e-45), df=np.float64(16150.0))
```

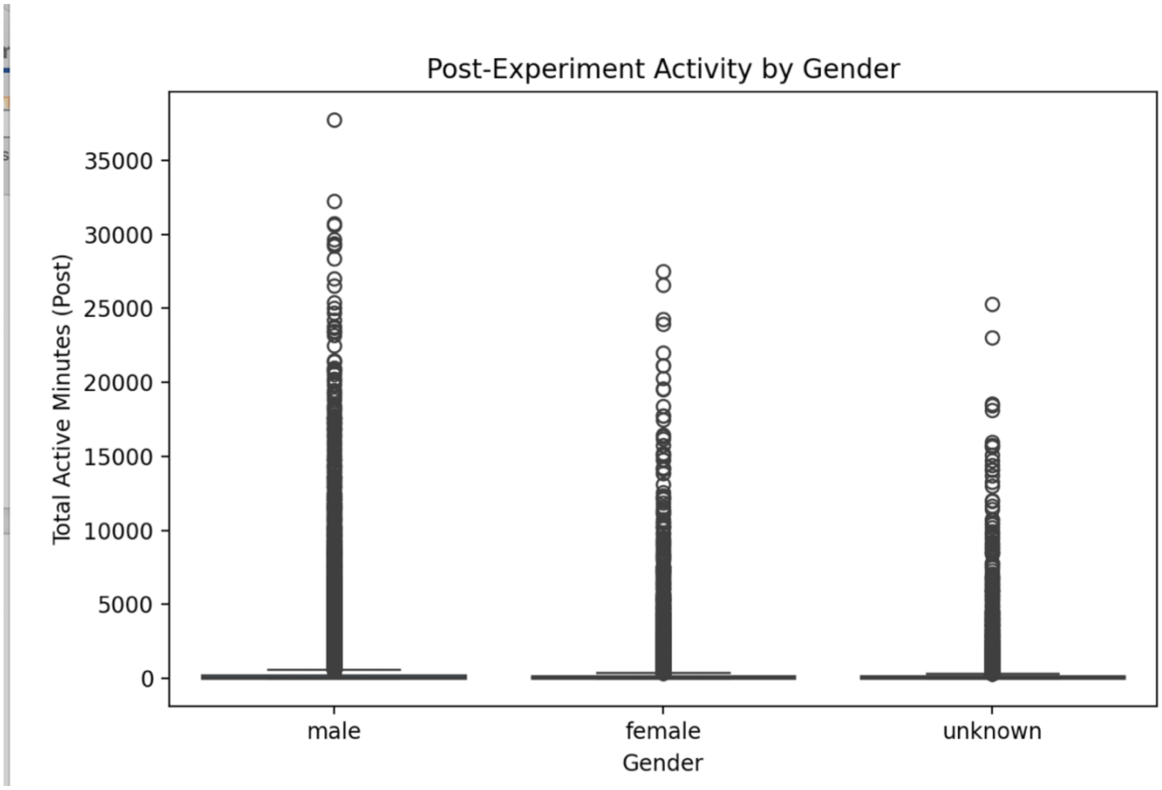
```
========================================
Control Group - Mean Pre: 96.53, Post: 86.75
Control Group - Median Pre: 46.00, Post: 39.00
========================================
Treatment Group - Scipy T-Test Result: TtestResult(statistic=np.float64(14.085969883667726), pvalue=np.float64(8.53067433170
4647e-45), df=np.float64(16150.0))
========================================
Treatment Group - Pingouin T-Test Result:
                T          dof alternative         p-val          CI95%   cohen-d       BF10  power
T-test  14.08597  15368.896659   two-sided  8.797220e-45  [24.13, 31.93]  0.221668  1.081e+41    1.0
========================================
Treatment Group - Statsmodels T-Test Result: (np.float64(14.085969883667728), np.float64(8.530674331704281e-45), np.float64(
16150.0))
========================================
Treatment Group - Manual T-Test (NumPy): t-statistic = 14.0860
========================================
Treatment Group - ResearchPy T-Test Result:
                                   Independent t-test     results
0  Difference (total_act_mins_post - total_act_mi...      28.0255
1                             Degrees of freedom =   16150.0000
2                                              t =      14.0860
3                          Two side test p value =       0.0000
4                         Difference < 0 p value =       1.0000
5                         Difference > 0 p value =       0.0000
6                                      Cohen's d =       0.2217
7                                      Hedge's g =       0.2217
8                                  Glass's delta1 =       0.2002
9                                Point-Biserial r =       0.1102
========================================
Treatment Group - Mean Pre: 82.79, Post: 110.82
Treatment Group - Median Pre: 40.00, Post: 55.00
```

The new analysis reveals statistically significant differences between pre- and post-experiment engagement for both the control and treatment groups. The control group experienced a significant decrease in total active minutes (Mean Pre: 96.53 → Post: 86.75, p-value = 2.35×10^{-23}), suggesting that user engagement naturally declined over time. However, the effect size (Cohen's d = -0.0786) is small, indicating that while the decline is statistically significant, its practical impact is minor. On the other hand, the treatment group showed a significant increase in engagement (Mean Pre: 82.79 → Post: 110.82, p-value = 8.53×10^{-45}), with a larger effect size (Cohen’s d = 0.2217), suggesting that the platform update successfully increased user engagement. Although the effect is not massive, it is statistically significant and demonstrates that the update helped counteract the natural decline in engagement seen in the control group.

There is a statistically significant difference between the control and treatment groups.

Part 6: Exploring other conclusions (10 points)






```
(DS5110) (base) ➜  part6 python 11.py

User Type Analysis:

non_reader: Pre-Mean = 105.93, Post-Mean = 116.62, T-Stat = 4.5590, P-Value = 0.0000
reader: Pre-Mean = 1632.32, Post-Mean = 1590.62, T-Stat = -1.0099, P-Value = 0.3126
new_user: Pre-Mean = 5.95, Post-Mean = 34.85, T-Stat = 9.7313, P-Value = 0.0000
contributor: Pre-Mean = 4594.98, Post-Mean = 4214.87, T-Stat = -1.4635, P-Value = 0.1435

Gender Analysis:

male: Pre-Mean = 544.04, Post-Mean = 543.65, T-Stat = -0.0244, P-Value = 0.9805
female: Pre-Mean = 351.72, Post-Mean = 344.76, T-Stat = -0.4302, P-Value = 0.6671
unknown: Pre-Mean = 348.79, Post-Mean = 326.08, T-Stat = -1.0333, P-Value = 0.3015
2025-02-03 11:18:36.301 python[70475:41033363] +[IMKClient subclass]: chose IMKClient_Modern
2025-02-03 11:18:36.301 python[70475:41033363] +[IMKInputSession subclass]: chose IMKInputSession_Modern
Analysis and visualization complete. Insights can be derived from the boxplots and statistical results.
```

The analysis of user types reveals key behavioral differences in response to the platform update. New users showed a significant increase in engagement (Pre-Mean: 5.95 → Post-Mean: 34.85, $p < 0.001$), indicating that the update successfully improved retention and activity for new users. Similarly, non-readers also experienced a significant increase in activity (Pre-Mean: 105.93 → Post-Mean: 116.62, $p < 0.001$), suggesting that the update may have introduced features that encouraged previously inactive users to spend more time on the platform. However, for readers and contributors, engagement either remained unchanged or declined slightly ($p > 0.05$), indicating that more engaged users were not significantly affected by the update. This suggests that while the update was effective in attracting and engaging new users, it did not enhance the experience for highly active users, which could indicate the need for additional improvements tailored to this segment.

In terms of gender analysis, the results indicate no statistically significant differences between pre- and post-experiment engagement across male, female, and unknown gender users (all p-values > 0.05). The update appears to have had a neutral effect across gender demographics, meaning no specific group benefitted disproportionately. This suggests that user engagement changes were driven more by usage behavior (e.g., new users vs. contributors) rather than gender-based preferences. These findings indicate that future platform improvements should focus on providing more value to existing engaged users, while maintaining the positive trend of attracting and retaining new users.

The code was stored in the 11.py in the part6 folder, and also the reorganized data (t1_t2_t3_t4.csv) was stored in the part6 folder.

Part 7: Summarize Your Results (10 Points)

Part 1: Getting to Know Your Data

This part involved understanding the structure and purpose of the provided datasets. It helped clarify that t1 and t3 contained user activity data before and after the experiment, while t2 provided group assignment, and t4 contained user demographics. This initial step was essential for determining how to process and analyze the data effectively in later parts.

Part 2: Organizing the Data

The data was cleaned and aggregated to facilitate meaningful comparisons. User activity minutes were summed across all dates, and each user was linked to their respective experiment group. This transformation allowed for a straightforward statistical analysis in the next phase, ensuring that the dataset was structured optimally for hypothesis testing.

Part 3: Statistical Analysis

A t-test was conducted to compare the total active minutes between the control and treatment groups. The results showed no statistically significant difference, suggesting that the new platform update did not have a measurable impact on engagement. Additionally, calculating the mean and median reinforced this conclusion, as both groups exhibited similar average activity levels.

Part 4: Digging a Little Deeper

Further investigation revealed that the data was highly skewed, with many extreme outliers. A box plot confirmed the presence of these outliers, some of which were unrealistically high (e.g., 99,999 minutes). This led to the decision to remove any data points exceeding a reasonable threshold (e.g., 12 hours per day). After cleaning the data, a reanalysis was performed, which still found no significant effect of the update.

Part 5: Digging Even Deeper

By incorporating t3, a before-and-after comparison was possible. The control group experienced a natural decline in engagement over time, while the treatment group showed a statistically significant increase. This suggests that the update may have successfully counteracted user attrition, leading to a meaningful improvement in engagement. And finally, we find that there is a statistically significant difference between the control and treatment groups.

Part 6: Exploring Other Conclusions

Examining demographic data revealed that new users and previously inactive users showed the most improvement in engagement after the update, while more active users (readers and contributors) experienced little change. Gender analysis showed no significant differences in engagement, indicating that the update's impact was more dependent on user type than gender.