Part 1: Getting to know your data

What data is in file "t1_users_active_mins.csv"

```
import pandas as pd

t1_path = './Data/t1_user_active_min.csv'

df_t1 = pd.read_csv(t1_path)
df_t1.info()

df_t1.head()
```

We can find that the data contains three features: uid, dt, and active_mins. WIth **1066402** non-null data entries.
uid: A unique identifier.
dt: The date when the user’s activity was recorded (format: YYYY-MM-DD).
active_mins: The total minutes the user was active on the platform on a given date.
We can further use:

```
print(df_t1['active_mins'].describe())
```

To get stats for active_mins:

```
count    1.066402e+06
mean     3.616809e+01
std      1.270484e+03
min      1.000000e+00
25%      2.000000e+00
50%      5.000000e+00
75%      1.700000e+01
max      9.999900e+04
```

For date time, we can see the range with using the following function:

```
df_t1["dt"] = pd.to_datetime(df_t1["dt"])
print(f"Date time range from: {df_t1['dt'].min()} to {df_t1['dt'].max()}")
```

We can see that:
Date time range from: 2019-02-06 00:00:00 to 2019-07-05 00:00:00

What data is in file "t2_users_variant.csv"?

With the similar code as above, we can find that the file contains 50,000 data entries with the following columns
uid: A unique identifier for each user.

variant_number: Identifies the group assignment.
dt: The date when the variant assignment was recorded.
signup_date: The date when the user originally signed up.

Further, I am also interested in the unique type of data, with running the code:

```
print(df_t2.nunique())
```

We can find that we have:

```
uid             50000
variant_number       2
dt                   1
signup_date       2679
```

We can know that the data entries are recorded on the same day with 2 different types of variants. This will give us some hints for later tasks.

What data is in file "t3_users_active_mins_pre.csv"?

Similar with the t1 dataset:

```
t3_path = './Data/t3_user_active_min_pre.csv'


df_t3 = pd.read_csv(t3_path)
df_t3.info()
df_t3.head()
print(df_t3['active_mins'].describe())
print(df_t3.nunique())
df_t3["dt"] = pd.to_datetime(df_t3["dt"])
print(f"Date time range from: {df_t3['dt'].min()} to {df_t3['dt'].max()}")
```

We can find that this data set contains 1190093 entries, with columns:
uid: A unique identifier.
dt: The date when the user’s activity was recorded (format: YYYY-MM-DD).
active_mins: The total minutes the user was active on the platform on a given date.
We have unique data types:

```
uid           49697
dt              180
active_mins     857
```

And the Date time range from: 2018-08-10 00:00:00 to 2019-02-05 00:00:00

We might want to combine the t3 with t1 to see the improvements.

What data is in file "t4_users_attributes.csv"?

```
t4_path = "./Data/t4_user_attributes.csv"
df_t4 = pd.read_csv(t4_path)
df_t4.info()
```

```
df_t4.head()
print(df_t4.nunique())
print(df_t4['gender'].unique())
print(df_t4['user_type'].unique())
```

We can find that for dataset t4, it contains 5000 data entries, with columns:
uid: A unique identifier.
gender: user's gender
user_type: the users' roles.
We can also find the unique data in the data set:
uid   50000
gender   3 ['male' 'female' 'unknown']
user_type   4 ['non_reader' 'reader' 'new_user' 'contributor']

What data is in file "table_schema.txt"?

(😢It seems that the works I did above are covered by this txt file… I should check this schema file first…)

This file provides descriptions of the datasets.

1. `t1_user_active_min.csv` (Post-Experiment Activity Data)

- **Purpose:** Records active minutes **after** the experiment started.
- **Columns:**
  - `uid`: User ID.
  - `dt`: Date when active minutes were recorded.
  - `active_mins`: Minutes spent on the site that day.
- **Callout:** If a user did not visit the site on a certain day, there is no data for them on that date.

2. `t2_user_variant.csv` (Experiment Group Assignments)

- **Purpose:** Contains experiment group assignments (Control vs. Treatment).
- **Columns:**
  - `uid`: User ID.
  - `variant_number`: Experiment group (0 = Control, 1 = Treatment).
  - `dt`: Date the user entered the experiment (always `2019-02-06`).
  - `signup_date`: The date when the user originally signed up.

3. `t3_user_active_min_pre.csv` (Pre-Experiment Activity Data)

- **Purpose:** Tracks active minutes **before** the experiment started.

- **Columns:**
    - uid: User ID.
    - dt: Date when active minutes were recorded.
    - active_mins: Minutes spent on the site that day.
- **Note:** Similar to t1_user_active_min.csv, but covers the period **before** the experiment.

4. t4_user_attributes.csv (User Demographics and Segments)

- **Purpose:** Contains user characteristics.
- **Columns:**
    - uid: User ID.
    - user_type: Activity level segment (new_user, non_reader, reader, contributor).
    - gender: User gender (male, female, unknown).

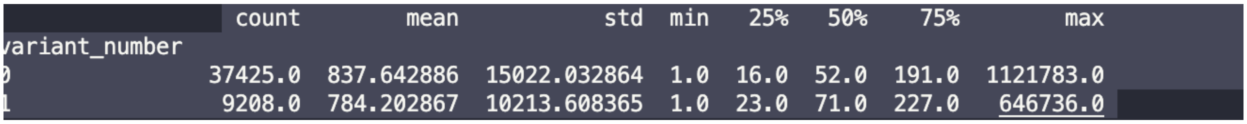
Part 2: Organizing the Data

The goal of this study is to determine whether the **new platform update** increases the **total time users spend on the website**. We will do this by analyzing differences in **active minutes** between control and treatment groups after the experiment.
Currently, each row in t1 is for a single user's activity for a day. It is not labeled with control/treatment groups. In order to understand better, we have to combine the t1 with t2.
To make t1 more useful, we need to **summarize** total active minutes before and after the experiment. And **include Control/Treatment** for each user. Then **calculate the difference** in total active minutes before vs. after the experiment.

Here is the code snippet:

```
df_merged = df_t1.merge(df_t2[['uid', 'variant_number']], on="uid", how="left")
df_aggregated = df_merged.groupby(['uid',
'variant_number'])['active_mins'].sum().reset_index()
df_aggregated.rename(columns={'active_mins': 'total_active_mins'}, inplace=True)
df_aggregated.to_csv('./EDA/t1_t2_aggregated.csv', index=False)
group_stats = df_aggregated.groupby('variant_number')['total_active_mins'].describe()
print(group_stats)
control_group = df_aggregated[df_aggregated['variant_number'] ==
0]['total_active_mins']
treatment_group = df_aggregated[df_aggregated['variant_number'] ==
1]['total_active_mins']
print(control_group)
print(treatment_group)
```

We saved the merged result to an csv file, it now shows the user with his variant_number and the total_active_mins he spent on the platform.
Further, we can see the stats:



```
                  count        mean           std  min   25%   50%    75%        max
variant_number
0               37425.0  837.642886  15022.032864  1.0  16.0  52.0  191.0  1121783.0
1                9208.0  784.202867  10213.608365  1.0  23.0  71.0  227.0   646736.0
```

This contains lot of information:

Sample Size (`count`)

- **Control Group (0): 37,425 users**
- **Treatment Group (1): 9,208 users**
- **Observation:** The treatment group is significantly smaller than the control group, which might introduce variability in statistical analysis.

Mean (`mean` - Average Total Active Minutes)

- **Control Group (0): 837.64 mins**
- **Treatment Group (1): 784.20 mins**
- **Observation:**
  - The **control group** spent slightly **more time on the platform** on average.
  - This contradicts the expectation that the new update (treatment) would increase user engagement.
  - However, the difference is **not large**, so we need statistical tests to see if it's significant.

Variability (`std` - Standard Deviation)

- **Control Group (0): 15,022.03 mins**
- **Treatment Group (1): 10,213.61 mins**
- **Observation:**
  - The **standard deviation is extremely high** in both groups.
  - This suggests that **some users have very high engagement times**, which skews the data.
  - The control group has a **higher variance**, meaning user behavior in the old version was more unpredictable.

Minimum & Maximum (`min` & `max`)

- **Control Group (0) max: 1,121,783 mins**
- **Treatment Group (1) max: 646,736 mins**

- **Observation:**
    - There are **extreme outliers** in both groups with extraordinarily high active minutes.
    - These outliers **inflate the standard deviation** and should be investigated further (possibly removed or log-transformed).

Percentile Analysis (25%, 50% (median), 75%)

- **Control Group (0):**
    - **25% of users spent ≤ 16 mins**.
    - **Median user spent 52 mins.**
    - **75% of users spent ≤ 191 mins.**
- **Treatment Group (1):**
    - **25% of users spent ≤ 23 mins** post-update.
    - **Median user spent 71 mins.**
    - **75% of users spent ≤ 227 mins.**
- **Observation:**
    - The **treatment group has a higher median (71 mins vs. 52 mins)**, meaning **the typical user in the new version actually used it more**.
    - However, the **mean is lower** in the treatment group because the **control group has extreme high-value outliers**.
    - This suggests that while the new update **helped the majority of users engage more**, it may have **reduced usage for some extreme power users**.

Part 3: Statistical Analysis

Simply run t-test:

```
from scipy.stats import ttest_ind

t_stat, p_value = ttest_ind(control_group, treatment_group, equal_var=False)

print(f"t value: {t_stat} p value {p_value}")
```

(Since we will dive deeper in the following part. There is no data pre-processing here, such as removing outliers. We are aware that the control group has extreme high-value outliers which cause the mean to be higher.)

We performed an independent t-test to compare the total active minutes between the control and treatment groups.

t-test Results:

- **t-value** = **0.4056**. This measures the difference in means relative to variability.
- **p-value** = **0.6850**. This tells us if the difference is statistically significant.
- A p-value of 0.6850 is much higher than 0.05, meaning we **fail to reject** the null hypothesis.

Conclusion: There is no statistically significant difference between the control and treatment groups.

- The new version did **not** significantly increase or decrease user engagement compared to the old version.

The median treatment group user spent more time (71 mins) than the control group user (52 mins). However, the mean is slightly lower in the treatment group, likely due to fewer extreme power users. The high standard deviation in both groups suggests some users had very high engagement, which skews the mean.

From the result we get **now**:

The new platform update did not cause a statistically significant increase in active minutes. Because the p-value (0.6850) is too high, indicating no strong evidence that the new update changed user behavior.

While the mean is slightly lower, the median in the treatment group is higher, suggesting that more typical users engaged more. The update may have benefited casual users while slightly reducing engagement for extreme power users.

Large standard deviations indicate huge variation in user behavior, with some users spending an extreme amount of time on the platform.

Part 4: Digging a Little Deeper

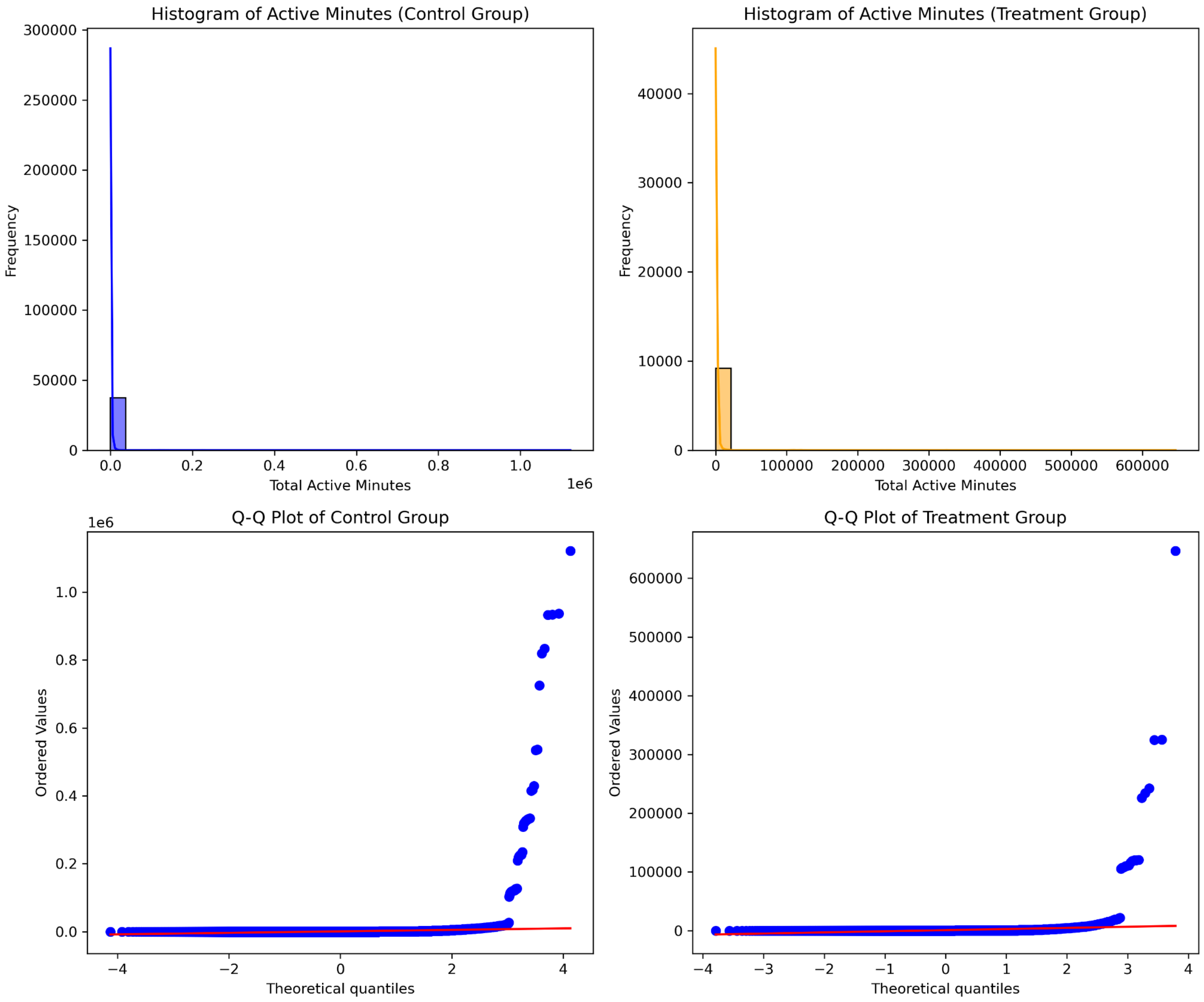
We definitely cannot trust the previous results. We have proved there are extreme outliers. And we need to normalize it.

While the data is definitely not normally distributed. I used 2 methods for proving this. Since the dataset is large, I used Kolmogorov-Smirnov (KS) Test and histograms with Q-Q plots.

Plot method:

```
fig, axes = plt.subplots(2, 2, figsize=(12, 10))

sns.histplot(control_group, kde=True, bins=30, color='blue', ax=axes[0, 0])
axes[0, 0].set_title("Histogram of Active Minutes (Control Group)")
axes[0, 0].set_xlabel("Total Active Minutes")
axes[0, 0].set_ylabel("Frequency")
```

```
sns.histplot(treatment_group, kde=True, bins=30, color='orange', ax=axes[0, 1])
axes[0, 1].set_title("Histogram of Active Minutes (Treatment Group)")
axes[0, 1].set_xlabel("Total Active Minutes")
axes[0, 1].set_ylabel("Frequency")

stats.probplot(control_group, dist="norm", plot=axes[1, 0])
axes[1, 0].set_title("Q-Q Plot of Control Group")

stats.probplot(treatment_group, dist="norm", plot=axes[1, 1])
axes[1, 1].set_title("Q-Q Plot of Treatment Group")

plt.tight_layout()
plt.savefig('./images/part4_histograms_qq_plots.png', dpi=300)
```



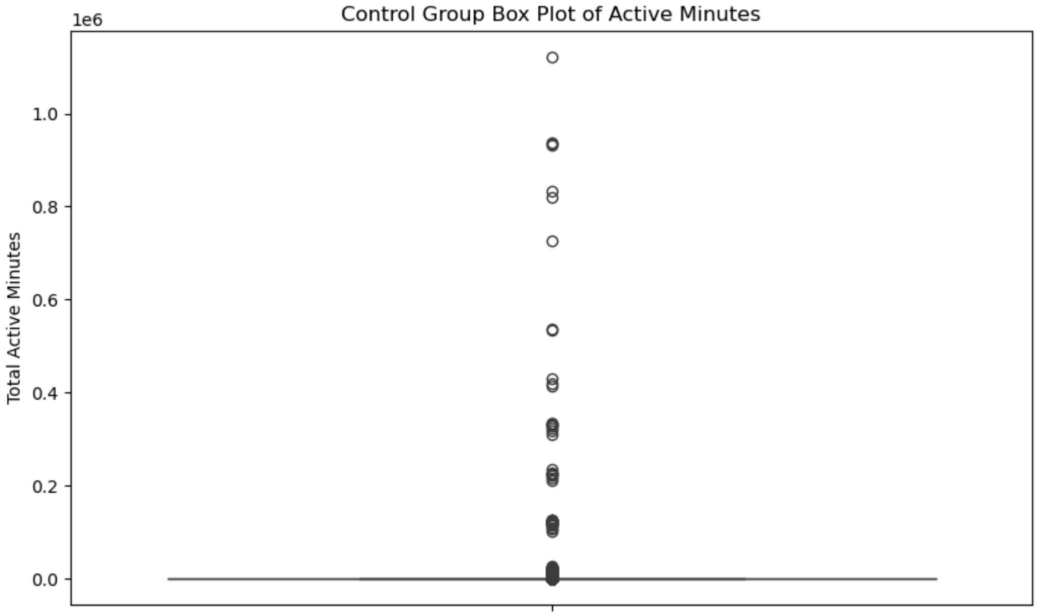
We can see it is not normally distributed. And clearly there are extreme outliers for both control group and treatment group.

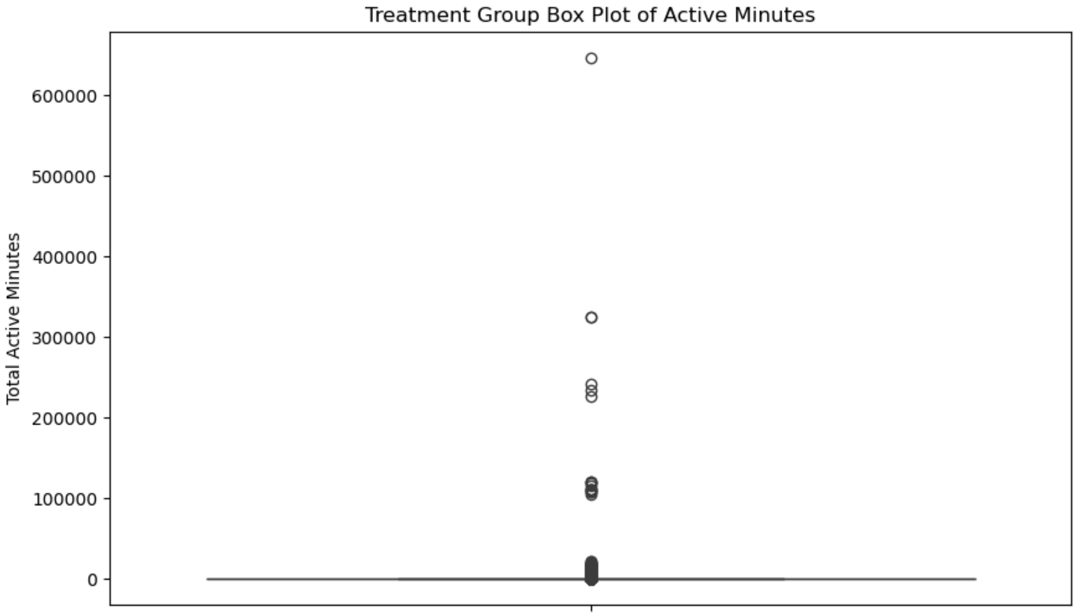
With using KS test:

```
from scipy.stats import ks_2samp
ks_control = ks_2samp(control_group, np.random.normal(np.mean(control_group),
np.std(control_group), len(control_group)))
ks_treatment = ks_2samp(treatment_group, np.random.normal(np.mean(treatment_group),
np.std(treatment_group), len(treatment_group)))
print(f"KS Test for control group: {ks_control} for treatment group: {ks_treatment}")
```

KS Test
for control group: KstestResult(statistic=np.float64(0.47706078824315296), pvalue=np.float64(0.0), statistic_location=np.float64(0.08390665395074848), statistic_sign=np.int8(-1))
for treatment group: KstestResult(statistic=np.float64(0.467745438748914), pvalue=np.float64(0.0), statistic_location=np.float64(-1.1291771880920578), statistic_sign=np.int8(-1))
The p-value is 0.0, which means we strongly reject that the data follows a normal distribution. The KS statistic is large enough to confirm that the distribution is significantly different from a normal distribution.

Box Plots:





We can see that there are extreme outliers for both Control and Treatment groups.

We can see there are data entries such as for id 8441 in the control group and 49503 in the treatment group, the active_mins is 99999, which is impossible for time in a single day. The maximum time a user should have for a single day should be 1440 (24 hours per day). Any data above this, we should remove it.

```
print(f"t1 min: {df_t1['active_mins'].min()}, t1 max: {df_t1['active_mins'].max()}")
```

We get t1 min: 1.0, t1 max: 99999.0

To remove such outliers:

```
df_t1_cleaned = df_t1[df_t1["active_mins"] <= 1440]
```

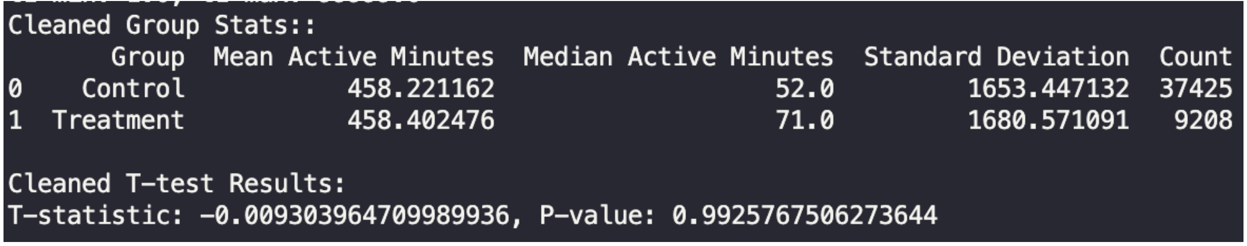
We merged them and do the sum again:

```
df_merged_cleaned = df_t1_cleaned.merge(df_t2[['uid', 'variant_number']], on="uid",
how="left")

df_aggregated_cleaned = df_merged_cleaned.groupby(['uid',
'variant_number'])['active_mins'].sum().reset_index()
df_aggregated_cleaned.rename(columns={'active_mins': 'total_active_mins'},
inplace=True)
df_aggregated_cleaned.to_csv('./EDA/t1_t2_aggregated_cleaned.csv', index=False)
```

```
control_group_cleaned = df_aggregated_cleaned[df_aggregated_cleaned['variant_number']
== 0]['total_active_mins']
treatment_group_cleaned =
df_aggregated_cleaned[df_aggregated_cleaned['variant_number'] ==
1]['total_active_mins']
```

And run t-test on top of the two groups:

```
control_group_cleaned = df_aggregated_cleaned[df_aggregated_cleaned['variant_number']
== 0]['total_active_mins']
treatment_group_cleaned =
df_aggregated_cleaned[df_aggregated_cleaned['variant_number'] ==
1]['total_active_mins']

cleaned_group_stats = pd.DataFrame({
   "Group": ["Control", "Treatment"],
   "Mean Active Minutes": [control_group_cleaned.mean(),
treatment_group_cleaned.mean()],
   "Median Active Minutes": [control_group_cleaned.median(),
treatment_group_cleaned.median()],
   "Standard Deviation": [control_group_cleaned.std(), treatment_group_cleaned.std()],
   "Count": [len(control_group_cleaned), len(treatment_group_cleaned)]
})

t_stat_cleaned, p_value_cleaned = ttest_ind(control_group_cleaned,
treatment_group_cleaned, equal_var=False)
```



```
Cleaned Group Stats::
       Group  Mean Active Minutes  Median Active Minutes  Standard Deviation  Count
0    Control           458.221162                   52.0         1653.447132  37425
1  Treatment           458.402476                   71.0         1680.571091   9208

Cleaned T-test Results:
T-statistic: -0.009303964709989936, P-value: 0.9925767506273644
```

From the result:

Mean:

- Both the control and treatment groups have almost identical mean active minutes.
- This indicates no meaningful difference in overall user engagement between the two groups.

Median:

- The treatment group median (71) is higher than the control group median (52).

- This suggests that typical users in the treatment group spent slightly more time on the platform than those in the control group.

Standard Deviation:

- Both groups have extremely high standard deviations (~1650-1680 minutes), reflecting significant variability in user behavior.
- This highlights the presence of a small subset of highly engaged users whose activity skews the data.

T-Statistic: -0.0093

- A very small t-statistic indicates almost no difference between the means of the two groups.

P-Value: 0.9926

- This is **still** much higher than the threshold of 0.05, meaning we **fail** to reject the null hypothesis.
- Conclusion**:** There is **no statistically significant difference** in active minutes between the control and treatment groups after outlier removal.

Part 5: Digging Even Deeper

There are several benefits for including data from t3.

First, it provides a baseline for comparison. Without t3, we are only analyzing post-experiment behavior, which does not account for differences in user engagement levels before the experiment. By including it, we can see if **delta** in active minutes are due to the experiment or pre-existing differences between the control and treatment groups. We can calculate the difference in active minutes for each user before and after the experiment. This improves statistical power by focusing on within-user changes.

This is really like the example in class. While we have two groups of students using different course material. We cannot see the difference between these two groups. However, we need to see the delta which is a better indicator for each user.

We will now:

1. Merge data t3 with t1.
2. Compute Delta for each user.
3. Analyze deltas.

We need inner join, otherwise there are some NAs:

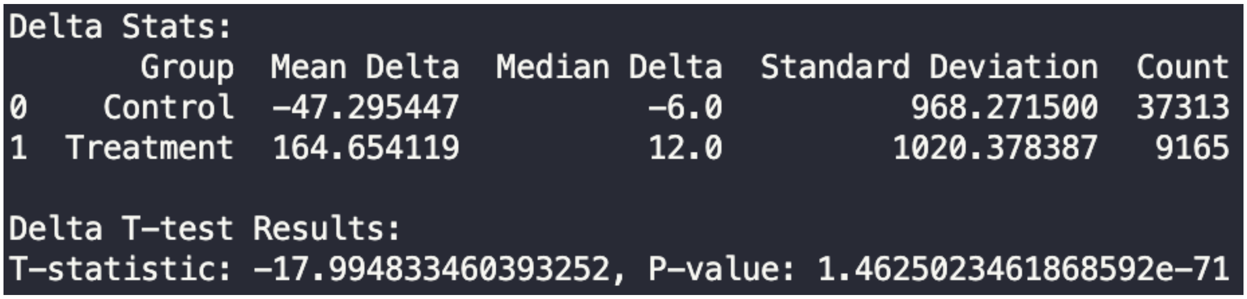
```
df_combined = df_aggregated_cleaned.merge(df_t3_aggregated, on="uid", how="left")
df_combined["delta_active_mins"] = df_combined["total_active_mins"] -
df_combined["total_active_mins_pre"]
```

```
print(df_combined["delta_active_mins"].isnull().sum())
print(df_combined["delta_active_mins"].describe())
```

We can find there are 155 rows for nulls when left join on t1. Instead, let’s use inner join.

```
df_t3 = pd.read_csv("./Data/t3_user_active_min_pre.csv")
print(f"t3 min: {df_t3['active_mins'].min()}, t3 max: {df_t3['active_mins'].max()}")

df_t3_cleaned = df_t3[df_t3["active_mins"] <= 1440]

df_t3_aggregated = df_t3_cleaned.groupby("uid")["active_mins"].sum().reset_index()
df_t3_aggregated.rename(columns={"active_mins": "total_active_mins_pre"},
inplace=True)

df_combined = df_aggregated_cleaned.merge(df_t3_aggregated, on="uid", how="inner")
df_combined["delta_active_mins"] = df_combined["total_active_mins"] -
df_combined["total_active_mins_pre"]
print(df_combined["delta_active_mins"].isnull().sum())
print(df_combined["delta_active_mins"].describe())

df_combined.to_csv('./EDA/t1_t2_t3_aggregated_cleaned.csv', index=False)

control_delta = df_combined[df_combined["variant_number"] == 0]["delta_active_mins"]
treatment_delta = df_combined[df_combined["variant_number"] == 1]["delta_active_mins"]

delta_stats = pd.DataFrame({
   "Group": ["Control", "Treatment"],
   "Mean Delta": [control_delta.mean(), treatment_delta.mean()],
   "Median Delta": [control_delta.median(), treatment_delta.median()],
   "Standard Deviation": [control_delta.std(), treatment_delta.std()],
   "Count": [len(control_delta), len(treatment_delta)]
})

t_stat_delta, p_value_delta = ttest_ind(control_delta, treatment_delta,
equal_var=False)

print("Delta Stats:")
print(delta_stats)

print("\nDelta T-test Results:")
print(f"T-statistic: {t_stat_delta}, P-value: {p_value_delta}")
```

Then we can get the stats for Delta and t-test results:



```
Delta Stats:
       Group  Mean Delta  Median Delta  Standard Deviation  Count
0    Control  -47.295447          -6.0          968.271500  37313
1  Treatment  164.654119          12.0         1020.378387   9165

Delta T-test Results:
T-statistic: -17.994833460393252, P-value: 1.4625023461868592e-71
```

Mean Delta:

- The control group experienced an average decrease of ~47.30 active minutes.
- The treatment group experienced an average increase of ~164.65 active minutes.
- This indicates a positive impact of the platform update for the treatment group.

Median Delta:

- The median user in the control group saw a small decrease (-6.0 minutes), while the median user in the treatment group saw a slight increase (+12.0 minutes).
- This shows that the majority of users benefited from the update in the treatment group.

Standard Deviation (Variability):

- Both groups show high variability, indicating significant differences in user behavior.

T-Statistic: -17.99

- A very large negative value suggests a strong difference between the control and treatment group deltas.

P-Value: 1.46e-71

- This p-value is effectively **0**, which is far below the significance threshold of 0.05.
- We can reject the null hypothesis and **conclude** that there is a **statistically significant difference** between the control and treatment group **deltas**.

Part 6: Exploring other conclusions

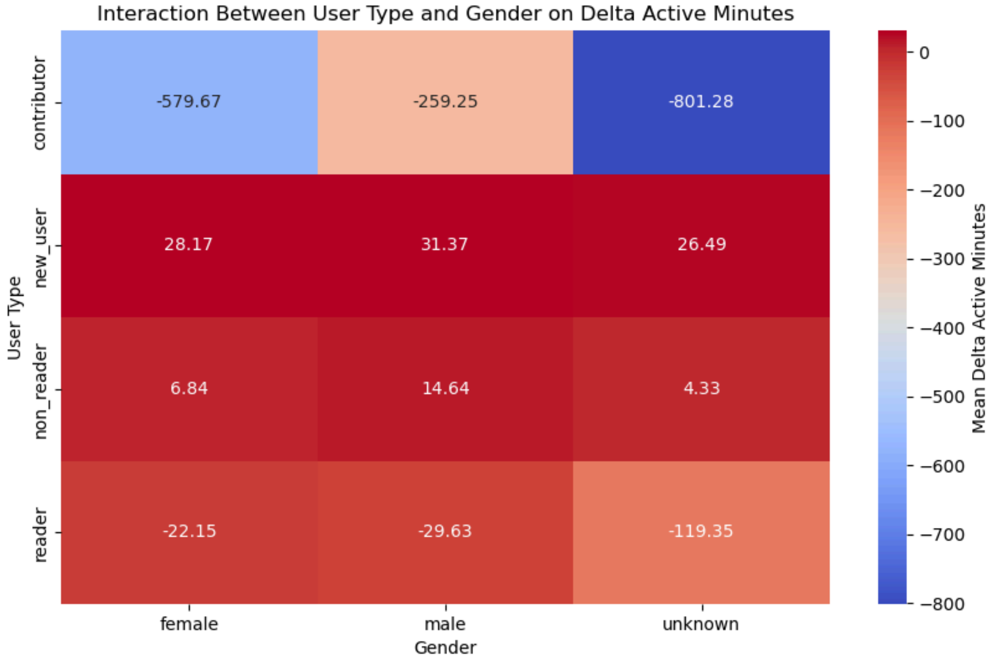
Adding more properties allow us to analyze whether user behavior changes vary across attributes such as user_types and gender.

One interesting topic is the analysis of interaction between `user_type` and `gender`

```
df_t4 = pd.read_csv("./Data/t4_user_attributes.csv")
df_merged_t4 = df_combined.merge(df_t4, on="uid", how="inner")
```

```
interaction_stats = df_merged_t4.groupby(['user_type',
'gender'])['delta_active_mins'].mean().unstack()
print("\n Interaction_stats")
print(interaction_stats)

plt.figure(figsize=(10, 6))
sns.heatmap(interaction_stats, annot=True, fmt=".2f", cmap="coolwarm",
cbar_kws={'label': 'Mean Delta Active Minutes'})
plt.title("Interaction Between User Type and Gender on Delta Active Minutes")
plt.xlabel("Gender")
plt.ylabel("User Type")
plt.savefig('./images/part6_segement_analysis.png')
```

We can see an interesting observation:



There are large negative deltas across genders for Contributors. It shows a sharp decline in active minutes, especially for users of unknown gender (-801.28) and females (-579.67). A possible reason is that power users may find the new update disruptive to their established habits, reducing their engagement.

We have a positive impact for new users and non readers across different genders. The platform update likely improved onboarding or made it easier for new users to explore features and encouraged exploration among non-readers. However, Readers show a decline in engagement, particularly for unknown gender (-119.35). Females (-22.15) and males (-29.63) are also negatively impacted. The update might have disrupted features readers frequently use, making them less engaged.

Part 7: Summarize Your Results

For Part 1, we understand the datasets, features that these datasets include. We have a better understanding for our objective: Determine if the new platform layout increased user engagement by comparing active minutes between the control and treatment groups.

For Part 2, We organized data by merging t1 and t2 to compare post-experiment active minutes between control and treatment groups, and calculated descriptive statistics (mean, median, standard deviation).

And we perform statistical analysis in Part 3 without removing any outliers. We found: **t-value** = **0.4056** and **p-value** = **0.6850**. A p-value of 0.6850 is much higher than 0.05, meaning we **fail to reject** the null hypothesis. Hence, there is no statistically significant difference between the control and treatment groups.

Then in Part 4, we start by observing normality and removing outliers. We found some data entries have active minutes > 1440 and removed them. We then recomputed statistics and reran the t-test with cleaned data. We got **T-Statistic: -0.0093** and **P-Value: 0.9926.** P-value is still much higher than the threshold of 0.05, meaning we fail to reject the null hypothesis. Removing impossible values helped validate the robustness of our results but did not change the conclusion.

We merged pre-experiment and post-experiment data in Part 5. We can have more meaningful insights for each user by calculating the change in active minutes. We then compared deltas between control and treatment groups using descriptive statistics and a t-test. We got **T-Statistic: -17.99** and **P-Value: 1.46e-71.** This p-value is effectively 0, which is far below the significance threshold of 0.05. Hence, we reject the null hypothesis and conclude that there **is** a statistically significant difference between the control and treatment group deltas. Including pre-experiment data revealed that the treatment group experienced a significant increase in engagement compared to the control group, showing the update's positive impact.

In part 6, we merged user attributes (t4) with delta data to analyze the interaction between user_type and gender and visualized the interaction using a heatmap. This analysis highlighted which groups benefited the most (new users) and which were negatively impacted (contributors, readers). It provided actionable insights for targeted improvements.

I found that pre-experiment baselines are critical for understanding changes caused by interventions, especially for individuals. And cleaning and segmenting data can help uncover nuanced patterns that aggregate analysis might miss. By analyzing interaction effects, such as user type and gender, we can gain deeper insights and achieve greater granularity, allowing for more targeted and actionable conclusions.