Bayesian A/B Testing

By Handi (Andy) Xie, Namika Takada, Chrstine Sangphet, Ann Liang, Cann Erozer

1. Introduction

2. Background

3. Related Work

4. Methodology

The goal of the following analyses is to evaluate the effectiveness of A/B testing in marketing settings. We employ both frequentist and Bayesian approaches to highlight the differences in their interpretations and outputs. Under the frequentist framework, chi-square tests analyze differences in observed frequencies between the two test groups for categorical distributions. Additionally, t-tests compare mean conversion rates between the two groups and calculate the t-statistic and p-value to determine whether the difference in conversion rates is statistically significant. Both frequentist approaches provide p-values to detect statistically significant differences; however, they lack the considerations of prior knowledge or uncertainty beyond the observed data.

Therefore, we adopt a Bayesian approach to address these limitations. This method incorporates prior beliefs and creates posterior distributions to quantify uncertainty and provide more interpretable results. It also enables posterior predictive checks, confidence intervals, and model selection where necessary.

4.1 Frequentist Approch

4.1.1 Dataset Overview

The dataset used for our analysis is the “Marketing A/B Testing” dataset, available on Kaggle, which provides insights into a marketing campaign aimed at increasing conversions through A/B testing. Marketing companies often seek to determine whether their campaigns will be successful and, more importantly, how much of that success can be attributed to the

advertisements themselves. In this case, the campaign was designed with a randomized control trial, dividing participants into two groups: the experimental group, which was exposed to advertisements, and the control group, which was either exposed to a public service announcement (PSA) or no ads at all. The analysis aims to identify factors that influence conversion rates, such as test group assignment and the timing of ads, to understand what drives success in the campaign.

The dataset includes key variables such as whether a user converted (i.e., bought the product), which we used as the target variable for our analysis. Additionally, it provides a user id for unique identification, the test group variable indicating whether a user saw the ad ("ad") or the PSA ("psa"), and information on the total ads seen by each user, as well as the most ads day and most ads hour, which indicate the day and hour during which the user encountered the most ads. By exploring these factors, we can gain insights into how ad timing and group assignment contribute to conversion outcomes.

4.1.2 Chi-Sqaure Test

A Chi-Square test is an appropriate statistical method for analyzing the relationship between categorical variables, as it assesses whether observed frequencies significantly differ from what we would expect under the null hypothesis. In our project, the test helps determine whether factors such as test group assignment (e.g., A (ad) vs. B (psa)), the most active day for ads (most ads day), or the most active hour for ads (most ads hour) are significantly associated with conversion rates. By evaluating these categorical variables against the “converted” outcome, the Chi-Square test enables us to identify whether variations in conversion rates across these categories are due to random chance or suggest underlying patterns. Specifically, our null hypothesis assumes no significant difference in conversion rates across groups. Rejecting this hypothesis indicates that the test group, day, or hour has a significant effect on conversion behavior.

To conduct this analysis, we used Python's *scipy.stats.chi2_contingency* function to perform the Chi-Square test for independence on each categorical variable. For each variable, we created a contingency table to capture the frequency distribution of its categories relative to the "Converted" outcome, which allowed us to calculate the Chi-Square test statistic and the p-value. Using a significance level of alpha = 0.05, we found that all three tests—test group assignment, most active ad day, and most active ad hour—produced extremely small p-values, well below the threshold. As a result, we rejected the null hypothesis for all variables, concluding that the observed differences in conversion rates are statistically significant and unlikely to be due to random variability. The results are as follows:

Statistical Significance: All three tests show statistically significant results, with p-values well below the significance threshold of 0.05, indicating that the differences in conversion rates are not due to random variability.

Key Findings: The analysis suggests that conversion rates are significantly influenced by test group assignment, the day with the most ads shown, and the hour with the most ads shown.

These results indicate that both time-related factors and group assignment play a significant role in conversion outcomes.

Sample Size: With a sample size of 588,101, the results are highly reliable and have greater statistical power. However, the large sample size could also detect very small, possibly insignificant differences as statistically significant, which may lead to overfitting or misinterpretation of the findings.

The figure below presents the code used for the test along with the corresponding results.

```
[ ]  from scipy.stats import chi2_contingency

    alpha = 0.05  # Significance level for the chi-squared test

    for variable in df_cat.columns:
        # Skip the target variable 'converted' to avoid self-comparison
        if variable != 'converted':
            # Create a contingency table (cross-tabulation) between the categorical variable and 'converted'
            contingency_table = pd.crosstab(df_cat[variable], df_cat['converted'])

            # Calculate the total sample size from the contingency table
            sample_size = np.sum(contingency_table.values)

            # Perform the chi-squared test for independence
            chi2, p, _, _ = chi2_contingency(contingency_table)

            # Display the chi-squared test results
            print(f"\nChi-Squared test for {variable} vs. converted:")
            print(f"Chi-squared value: {chi2:.4f}")  # Print the test statistic rounded to 4 decimal places
            print(f"p-value: {p:.4e}")  # Print the p-value in scientific notation for precision

            # Check if the p-value indicates a statistically significant result
            if p < alpha:
                print(f"The difference in conversion rates across {variable} is statistically significant.")
                # Reject the null hypothesis: there is a significant difference in conversion rates across groups
            else:
                print(f"There is no significant difference in conversion rates across {variable}.")
                # Fail to reject the null hypothesis: conversion rates are not significantly different across groups

    # Print the total sample size with a preceding blank line for clarity
    print(f"\nSample Size: {sample_size}")
```

```
Chi-Squared test for test group vs. converted:
Chi-squared value: 54.0058
p-value: 1.9990e-13
The difference in conversion rates across test group is statistically significant.

Chi-Squared test for most ads day vs. converted:
Chi-squared value: 410.0479
p-value: 1.9322e-85
The difference in conversion rates across most ads day is statistically significant.

Chi-Squared test for most ads hour vs. converted:
Chi-squared value: 430.7687
p-value: 8.0276e-77
The difference in conversion rates across most ads hour is statistically significant.

Sample Size: 588101
```

Figure 4.1.2: Code and results of the Chi-Square test for independence, evaluating the relationship between categorical variables and conversion rates.

4.1.3 T-Test

The t-test is used to compare the means of two independent test groups based on continuous data, as it helps to determine whether a statistically significant difference exists between the groups. In this project, we transformed the categorical value “Converted” into the continuous metric “Conversion Rate” by dividing the total conversions by the number of website visitors. The two test groups being analyzed are “ad” and “psa,” while other categorical factors “day of the week” and “hour of the day” are used as variables to explore conversion rate

differences. By conducting an independent two-sample t-test, we can assess whether the observed differences in conversion rates between the ad and psa groups are statistically significant. Here, the null hypothesis assumes that no significant difference exists between the two groups.

Using the ttest_ind function from the scipy.stats library, we performed the independent t-tests for each day of the week and each hour of the day. For the analysis on “day of the week,” the results indicate significant differences in conversion rates on Monday, Tuesday, Wednesday, Friday, and Saturday, which rejects the null hypothesis since the p-value is lower than 0.05. However, Thursday and Sunday show subtle differences with p-values is greater than 0.05, meaning the null hypothesis cannot be rejected. Similarly, for the analysis on “hour of the day,” certain timeframes, such as 10 AM to 2 PM, demonstrate significant differences in conversion rates, which leads to rejection of the null hypothesis. On the contrary, other timeframes, such as 3 PM to 4 PM, have no significant variation, and the null hypothesis cannot be rejected.

While the t-test provides valuable insights into statistical differences, it has limitations. The method is ideal for comparing only two groups and relies on key assumptions that the data should follow a normal distribution, the variances between groups should be approximately equal, and the groups must be independent. In this scenario, the conversion rates in the ad group must not influence those in the psa group. More seriously, the t-test focuses solely on rejecting or failing to reject the null hypothesis, without quantifying uncertainty in the results. These limitations emphasize the need for Bayesian approach that can incorporate prior information and provide a more comprehensive assessment of uncertainty through posterior distributions.

The figures below illustrates the codes for performing the t-tests and their associated outputs.

```python
from scipy.stats import ttest_ind

# Perform T-tests for each day and store results
ttest_results = []
unique_days = data['most ads day'].unique()

for day in unique_days:
    ad_group = data[(data['most ads day']==day) & (data['test group']=='ad')]['converted']
    psa_group = data[(data['most ads day']==day) & (data['test group']=='psa')]['converted']

    t_stat, p_val = ttest_ind(ad_group, psa_group, equal_var=False, nan_policy='omit')

    ttest_results.append({
        'Day': day,
        'T-statistic': round(t_stat, 4),
        'P-value': round(p_val, 4),
        'Significant': p_val < 0.05
    })

# Convert results to DataFrame
ttest_results_df = pd.DataFrame(ttest_results)

# Sort by day
weekday_order = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
ttest_results_df['Day'] = pd.Categorical(ttest_results_df['Day'], categories=weekday_order, ordered=True)
ttest_results_df = ttest_results_df.sort_values('Day').reset_index(drop=True)

# Display the results
ttest_results_df
```

	Day	T-statistic	P-value	Significant
0	Monday	4.1324	0.0000	True
1	Tuesday	6.9495	0.0000	True
2	Wednesday	4.3963	0.0000	True
3	Thursday	0.6084	0.5429	False
4	Friday	2.9161	0.0036	True
5	Saturday	3.2389	0.0012	True
6	Sunday	1.5336	0.1252	False

```python
from scipy.stats import ttest_ind

# Perform t-tests for each hour of the day
ttest_results_hour = []
unique_hours = data['most ads hour'].unique()

for hour in unique_hours:
    ad_group = data[(data['most ads hour']==hour) & (data['test group']=='ad')]['converted']
    psa_group = data[(data['most ads hour']==hour) & (data['test group']=='psa')]['converted']

    t_stat, p_val = ttest_ind(ad_group, psa_group, equal_var=False, nan_policy='omit')

    ttest_results_hour.append({
        'Hour': hour,
        'T-statistic': round(t_stat, 4),
        'P-value': round(p_val, 4),
        'Significant': p_val < 0.05
    })

# Convert results into a DataFrame
ttest_results_hour_df = pd.DataFrame(ttest_results_hour)

# Sort by hour
ttest_results_hour_df = ttest_results_hour_df.sort_values('Hour').reset_index(drop=True)

# Display the results
ttest_results_hour_df
```

	Hour	T-statistic	P-value	Significant
0	0	10.1970	0.0000	True
1	1	7.9266	0.0000	True
2	2	6.2682	0.0000	True
3	3	-0.0711	0.9435	False
4	4	3.3408	0.0009	True
5	5	4.0411	0.0001	True
6	6	6.8606	0.0000	True
7	7	1.6208	0.1062	False
8	8	2.2329	0.0258	True
9	9	2.8196	0.0049	True
10	10	2.7129	0.0067	True
11	11	2.8971	0.0038	True

	Hour	T-statistic	P-value	Significant
0	12	2.4331	0.0150	True
1	13	2.9848	0.0029	True
2	14	4.1542	0.0000	True
3	15	1.2464	0.2128	False
4	16	0.6719	0.5018	False
5	17	2.1283	0.0335	True
6	18	0.5104	0.6099	False
7	19	0.4994	0.6176	False
8	20	3.0494	0.0023	True
9	21	1.5187	0.1291	False
10	22	2.3428	0.0193	True
11	23	2.1521	0.0317	True

Figure 4.1.3: The codes and outputs associated with t-test that compares the mean conversion rates between the ad and psa groups based on day of the week and hour of the day.

4.2 Bayesian Approach

4.2.1 Misleading: P-values

P-values in frequentist A/B testing can be misleading for several reasons. Firstly, p-values are often misinterpreted as the probability that the null hypothesis is true, when they actually represent the probability of obtaining the results at least as extreme as those observed, assuming the null hypothesis is true. This subtle, but very significant, distinction leads to frequent misunderstanding of the true outcome in test results. P-values also do not provide information about the practical significance or effect size of an observed difference. A low p-value does not necessarily indicate a substantial impact on business metrics, in our case conversion rates, potentially leading to implementation of changes with minimal real-world benefit. The reliance on fixed significance thresholds ($p < 0.05$) can result in arbitrary decision-making. This can result in a miscommunication with stakeholders. A p value of 0.01 can be interpreted as many different things. This approach also fails to account for the continuous nature of evidence and can lead to an ill-timed conclusion.

4.2.2 Strengths of the Bayesian Approach

The Bayesian approach to A/B testing offers multiple advantages over the traditional frequentist method. One of the key benefits of Bayesian A/B testing is its flexibility and speed. Unlike the frequentist approach, which requires a predetermined sample size, the Bayesian methods allow for continuous updating of probabilities as new information is collected. This enables faster decision-making and more efficient experimentation for businesses to optimize their tests quickly. As mentioned previously, the nature of Bayesian results does not rely on the complex statistical concepts of p-values. The Bayesian approach provides a clear probability that one variant is outperforming another. This simplicity in interpretation is particularly valuable in business settings during the decision-making process.

The ability to incorporate prior knowledge is another strength of the Bayesian approach. By integrating historical/previous experiment results, the testing will exhibit a more well-rounded result and a comprehensive understanding of the probabilities presented. The

frequentist way phrases the problem to fit the model. This approach can be quite difficult and only provides you with the point estimate, confidence interval, and p value. However, the Bayesian approach finds a model that fits the problem. This gives you a posterior distribution that allows you to answer a wider range of questions for marketers, data scientists, and stakeholders.

The Bayesian approach can give a more nuanced understanding of experimental results. Bayesian A/B testing is a versatile tool for companies of all sizes and traffic to gather meaningful insights about their business. However, one drawback of the Bayesian approach is time. The Bayesian approach typically requires more computation power, especially for larger and more complex datasets. Inferences involving computing complex integrals over parameter spaces can be computation intensive, so this approach may require more power. Despite the computational challenges, the benefits from the Bayesian approach outweigh the costs.

5. Conclusion

6. Appendix

Code Repository

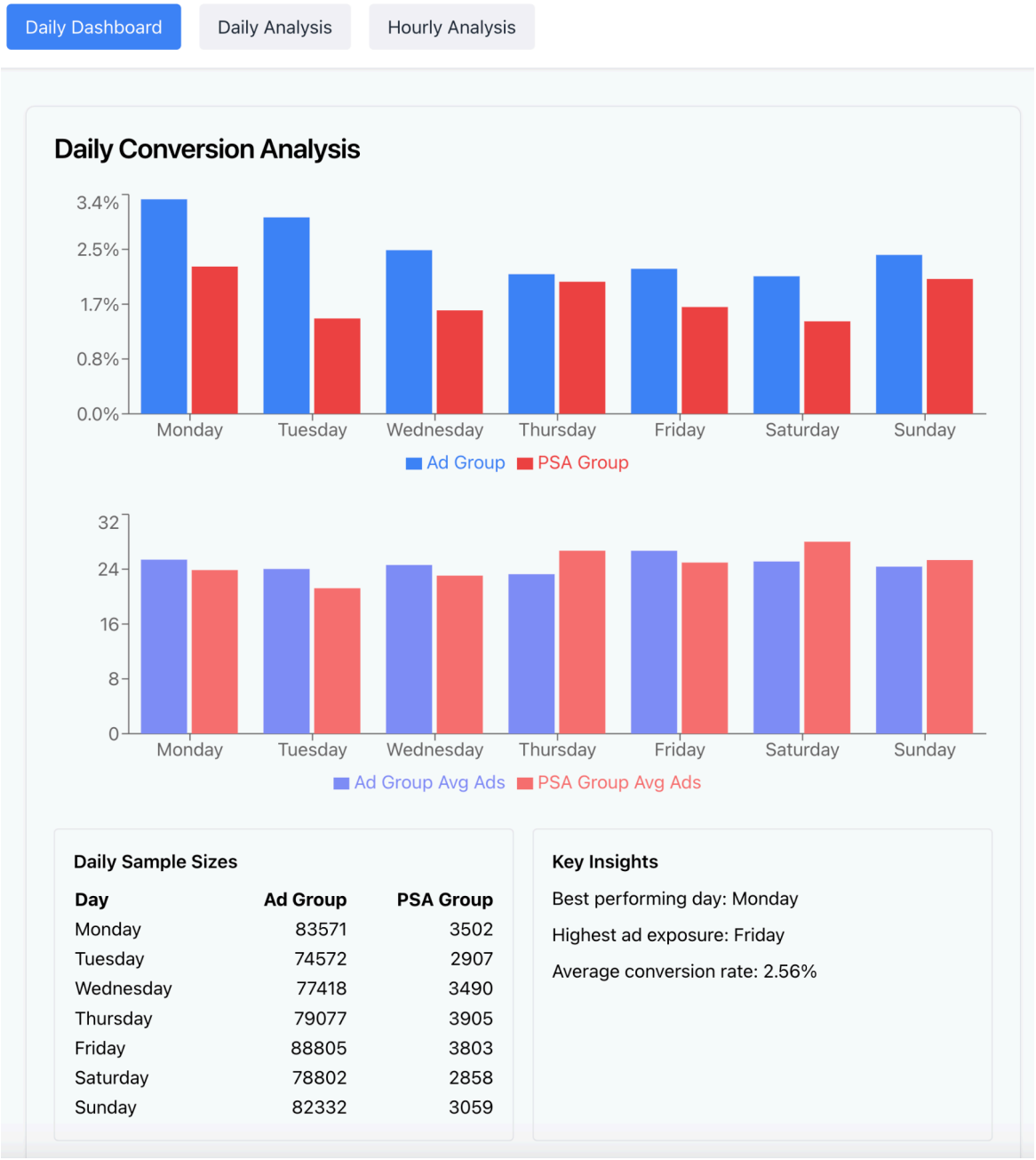
To reproduce the toolset we used for the demonstration, please go to our repository and refer to this README.md file https://github.com/PalmPalm7/bayesian_ab_testing for further understanding. The current dataset leverages the Kaggle dataset we used, a marketing dataset consists of two groups: PSA (public service announcement) and AD (advertisement).

To run the project, please

1. Make sure you have successfully installed NPM. (e.g. via https://nodejs.org/en/download/package-manager)
2. Clone the Github repository from the repo. (e.g. *git clone https://github.com/PalmPalm7/bayesian_ab_testing.git*)
3. Install the required npm packages. (e.g. *npm install jstat*)
4. Start running the React App by
    a. Using *npm start* or *npm restart*
    b. View the application on http://localhost:3000/, or the corresponding port.

To use for Bayesian A/B testing, please refer to the three specific dashboards.

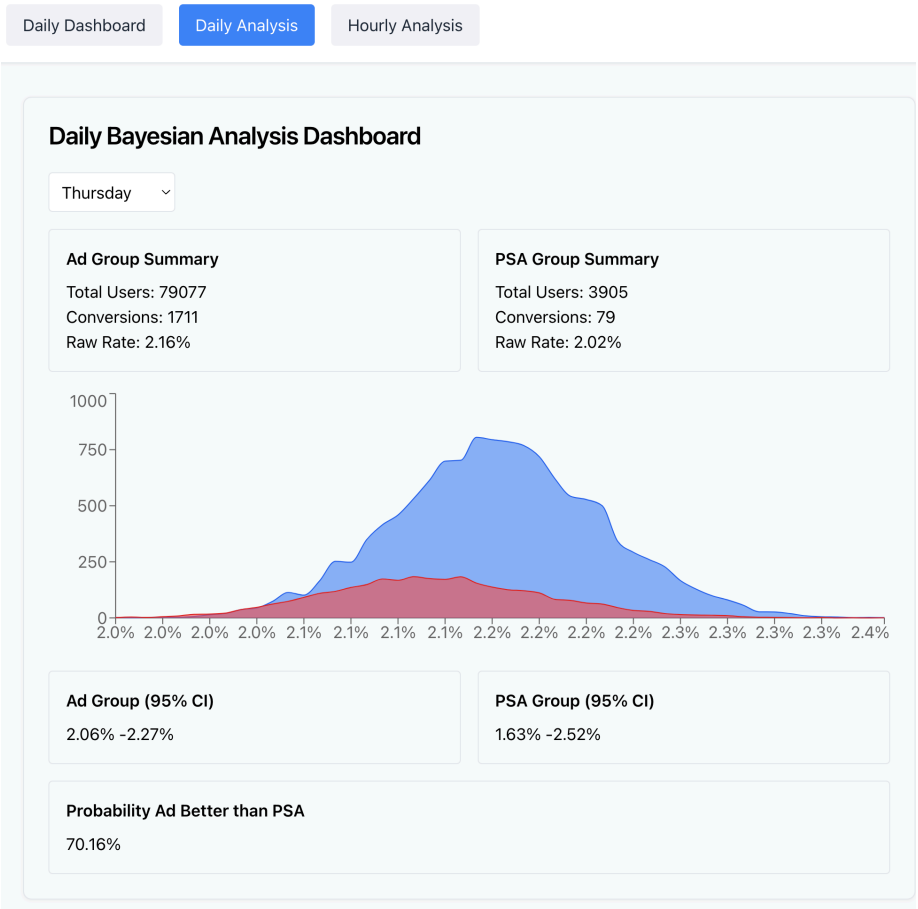
Daily Dashboard



The daily analysis dashboard is a summary of the whole dataset, with a visualization on the conversion rate and count of the advertisement impression per day.

The logic is self explanatory, for further explanation, please refer to: https://github.com/PalmPalm7/bayesian_ab_testing/blob/main/src/components/DailyDashboard.jsx

Daily Analysis Dashboard



This will be the main tool you will use to determine the effectiveness of the two groups.

The "Probability Ad Better than PSA" section calculates the fraction of posterior samples where the Ad group's sampled conversion rate is higher than the PSA group's sampled conversion rate. This fraction represents the Bayesian posterior probability that the Ad variant is better than the PSA variant, given the data and the chosen priors.

The posterior distribution is obtained by using a Beta distribution to model the conversion rate. Given some data (number of conversions and trials in each group), we update the prior Beta distributions for both groups to get posterior distributions. These posterior distributions reflect our updated beliefs about the true conversion rates after seeing the data.

For each group (Ad and PSA), the posterior is a Beta distribution determined by:

$$\textbf{Posterior}(p) = \textbf{Beta}(p;\ \alpha + \textbf{successes},\ \beta + \textbf{failures})$$

where α and β typically start at 1 and 1 (a uniform prior), and "successes" and "failures" are derived from the observed conversion data.

In the Javascript code, it is implemented this way:

```
const ad_alpha_post = 1 + adSuccesses;
const ad_beta_post = 1 + (adTrials - adSuccesses);
```

```
const psa_alpha_post = 1 + psaSuccesses;
const psa_beta_post = 1 + (psaTrials - psaSuccesses);
```

Then we draw SAMPLE_SIZE = 10,000 random samples from each posterior distribution where each sample represents a plausible "true" conversion rate scenario based on the observed data and the prior.

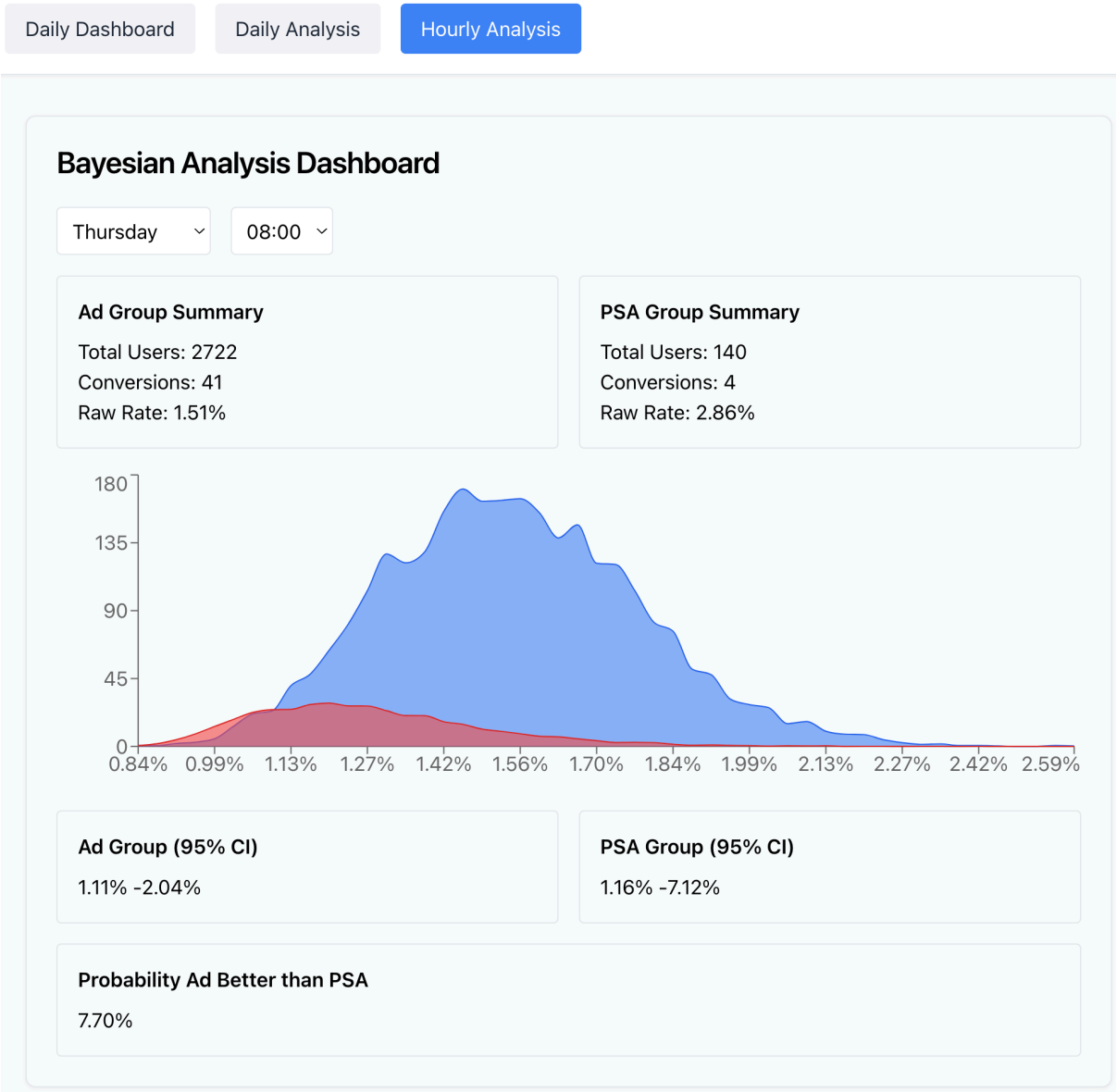
```
const adPosterior = Array.from({length: SAMPLE_SIZE}, () =>
jStat.beta.sample(ad_alpha_post, ad_beta_post));
const psaPosterior = Array.from({length: SAMPLE_SIZE}, () =>
jStat.beta.sample(psa_alpha_post, psa_beta_post));
```

In the end, the posterior comparison is done by estimating the probability that one group is better than the other, once we have arrays of samples from the Ad group's posterior and the PSA group's posterior.

```
let countAdBetter = 0;
for (let i = 0; i < SAMPLE_SIZE; i++) {
  if (adPosterior[i] > psaPosterior[i]) countAdBetter++;
}
const probAdBetter = countAdBetter / SAMPLE_SIZE;
```

In other words, for each pair of samples (adPosterior[i], psaPosterior[i]), it checks if adPosterior[i] > psaPosterior[i]. If this happens most of the time, it means that the Ad variant likely has a higher true conversion rate than the PSA variant, given the data and priors.

Hourly Analysis Dashboard



At last, there is a hourly analysis dashboard that utilizes the logic similar to the daily analysis dashboard.