Detecting the treatment effect based on simple statistical metrics

Xiao Chen[1]

[1]markcx@stanford.edu

1 Introduction

The motivation of our work is to implement simple statistical tools to detect the average treatment effects on grouped users. Taking advantages of a large number of users, companies, such as Microsoft Bing search, can roll out experiments to test if a new feature plays an important role or not by monitoring a number of key metrics. Many studies call such experiments A/B tests [4]. Our work introduces two major decision approaches, i.e. Bayes rule and Hypothesis testing, to support companies' business analysis or product delivery decisions.

2 Problem & Methodology

Let the data point 'sumX' be $X_{i,j,k}$, where the first subscription $i = 0, 1$ meaning that the control group is 0 and the treatment group is 1, the second subscription $j = 0, 1$ meaning that the pre-experiment is 0 and the post-experiment is 1, and the last subscription is the data points index in the corresponding group, specifically the $k = 1, \dots, N_C$ and $l = 1, \dots, N_T$. We summarize the denoted variables in Table 1.

Table 1: Collection of Variables

features		pre-exper	post-exper
sumX	ctrl	$X_{0,0,k}$	$X_{0,1,k}$
	treat	$X_{1,0,l}$	$X_{1,1,l}$
sumY	ctrl	$Y_{0,0,k}$	$Y_{0,1,k}$
	treat	$Y_{1,0,l}$	$Y_{1,1,l}$
PV	ctrl	$a_{0,0,k}$	$a_{0,1,k}$
	treat	$a_{1,0,l}$	$a_{1,1,l}$
numDays	ctrl	$d_{0,0,k}$	$d_{0,1,k}$
	treat	$d_{1,0,l}$	$d_{1,1,l}$
numVisits	ctrl	$v_{0,0,k}$	$v_{0,1,k}$
	treat	$v_{1,0,l}$	$v_{1,1,l}$

For simplicity, we will illustrate the calculation of 'sumX' as an example. Denote the feature 'sumX' as x, hence each observation is $x^{(j)}$ for $j = 1$ to $N_C + N_T$. Let z_1 be the binary variable indicating whether the sample is from the control group or the treatment group, z_2 be the indicator variable to represent if the sample is from pre-experiment or post-experiment. The feature 'sumX' can be expressed as

$$x^{(j)} = \beta_0 + \beta_1 z_1^{(j)} + \beta_2 z_2^{(j)} + \beta_3 z_1^{(j)} z_2^{(j)} + \varepsilon^{(j)} \qquad (1)$$

where $\varepsilon^{(j)}$ is normal distributed as $\mathcal{N}(0, \tau^2)$, and $\beta_0, \beta_1, \beta_2$ and β_3 are the corresponding coefficients.

To test whether there is a treatment effect or not, we essentially want to check whether β_3 is 0 or not. For the control pre-experiment group the indicator $z_1 = z_2 = 0$, thus $\mathbb{E}(X_{0,0,k}) = \beta_0$. For the control post-experiment group, the indicator $z_1 = 0, z_2 = 1$, $\mathbb{E}(X_{0,1,k}) = \beta_0 + \beta_2$. Similarly, we can derive that $\mathbb{E}(X_{1,0,l}) = \beta_0 + \beta_1$, $\mathbb{E}(X_{1,1,l}) = \beta_0 + \beta_1 + \beta_2 + \beta_3$. We approximate the expected value by the corresponding sample mean and drop the 3rd subscription k on $X_{i,j,k}$ to get the following relations

$$\overline{X}_{0,0} = \beta_0; \quad \overline{X}_{0,1} = \beta_0 + \beta_2; \quad \overline{X}_{1,0} = \beta_0 + \beta_1; \qquad (2)$$

$$\overline{X}_{1,1} = \beta_0 + \beta_1 + \beta_2 + \beta_3 \qquad (3)$$

Thus we have

$$\beta_3 = \overline{X}_{1,1} - \overline{X}_{0,1} - \overline{X}_{1,0} + \overline{X}_{0,0} \qquad (4)$$

$$= \underbrace{(\overline{X}_{1,1} - \overline{X}_{1,0})}_{\mu_{x,1}} - \underbrace{(\overline{X}_{0,1} - \overline{X}_{0,0})}_{\mu_{x,0}}. \qquad (5)$$

After we organize the data into groups, we can succinctly denote the result of $(\overline{X}_{1,1} - \overline{X}_{1,0})$ as $\mu_{x,1}$, and $(\overline{X}_{0,1} - \overline{X}_{0,0})$ as $\mu_{x,0}$, respectively. More extensively, we denote the mean of the corresponding features as follows

features	treatment	control
sumX	$\mu_{x,1}$	$\mu_{x,0}$
sumY	$\mu_{y,1}$	$\mu_{y,0}$
PV	$\mu_{a,1}$	$\mu_{a,0}$
numDays	$\mu_{d,1}$	$\mu_{d,0}$
numVisits	$\mu_{v,1}$	$\mu_{v,0}$

2.1 Bayes Approach

We will illustrate the calculation steps for feature 'sumX', and the other features share a similar logic flow. With the notation in [2], the pooled sample variance σ^2 satisfied the constraint that $\frac{\sigma^2}{N_E} = \frac{\sigma_C^2}{N_C} + \frac{\sigma_T^2}{N_T}$, where $\frac{1}{N_E} = \frac{1}{N_T} + \frac{1}{N_C}$. The $\Delta = \mu_{x,1} - \mu_{x,0}$. With the notion that $\delta = \frac{\Delta}{\sigma}$, the Z statistic can be expressed as

$$Z = \frac{\delta}{\sqrt{1/N_E}} = \frac{\Delta}{\sqrt{\sigma^2/N_E}} \sim \mathcal{N}(\theta_x, 1) \qquad (6)$$

$$\implies \mu_{x,1} - \mu_{x,0} \sim \mathcal{N}(\theta_x, \frac{\sigma^2}{N_E}) \qquad (7)$$

We set the standard hypothesis testing that $H_0 : \theta_x = 0; H_1 : \theta_x \neq 0$, where H_0 is the the null hypothesis and H_1

is the alternative. Let π_0 be the probability that the null hypothesis is true, namely $P(H_0) = P(\theta_x = 0) = \pi_0$, and α_0 to be the posterior probability that null hypothesis is true given the observed data, namely $P(\theta_x = 0|x) = \alpha_0$. We get the result that

$$\begin{aligned}\alpha_0 = P(H_0|x) &= P(\theta_x = 0|x) && (8)\\ &= \frac{f(x|\theta_x = 0)\pi_0}{f(x|\theta_x = 0)\pi_0 + f(x|\theta_x \neq 0)(1-\pi_0)} && (9)\\ &= \left(1 + \frac{1-\pi_0}{\pi_0}\frac{m_1(x)}{f(x|\theta_x = 0)}\right)^{-1}, && (10)\end{aligned}$$

where $m_1(x)$ is the marginal density. It also can be considered as the posterior distribution of θ_x.

In our experiment, we pick $\pi_0 = 0.9$, $f(x|\theta_x = 0) \sim \mathcal{N}(0, \sigma^2/N_E)$, and $m_1(x) \sim \mathcal{N}(0, V^2)$ where V^2 can be calculated as total sample variance. The rational of picking 0.9 as the prior probability is due to the belief that in many cases the metric/feature won't be changed significantly from pre-experiment to post-experiment. According to equation (10), the α_0 can be calculated numerically.

The Bayes's rule is to minimize the expected loss $\mathbb{E}^{\pi(\theta|x)}\big(L(\theta, \lambda)\big)$, where λ is the action. Define λ_i to be accepting hypothesis H_i, and the loss function is

$$L(\theta, \lambda_i) = \begin{cases} 0 & \text{if } H_i \text{ is true} \\ \zeta_i & \text{if } H_i \text{ is false} \end{cases}. \qquad (11)$$

The expected loss of taking action λ_0, namely accepting H_0, is calculated

$$\begin{aligned}\mathbb{E}^{\pi(\theta|x)}\big(L(\theta, \lambda_0)\big) &= \int_\theta L(\theta, \lambda_0)\pi(\theta|x)d\theta && (12)\\ &= \int_{H_1} \zeta_0 \pi(\theta|x)d\theta = \zeta_0 P(H_1|x) && (13)\end{aligned}$$

Similarly the expected loss for accepting H_1 is $\zeta_1 P(H_0|x)$. Thus the rule of accepting the null hypothesis is to check if the $\mathbb{E}^{\pi(\theta|x)}\big(L(\theta, \lambda_0)\big)$ is minimum, i.e. equivalently,

$$P(H_0|x) > \frac{\zeta_0}{\zeta_1} P(H_1|x) \implies \alpha_0 > \frac{\zeta_0}{\zeta_1}\alpha_1 \qquad (14)$$

Where α_1 is the posterior probability of an alternative hypothesis, namely $P(\theta_x \neq 0|x) = \alpha_1$. The penalty value of ζ_0 and ζ_1 can be varied by specific features. For simplicity we just set $\zeta_0 = 10$ and $\zeta_1 = 1$, which means delivering an insignificant feature will be punished less than delivering a significant feature.
Similar analysis can be applied to other features, we write the following expressions to illustrate this idea.

$$\begin{aligned}\mu_{y,1} - \mu_{y,0} &\sim \mathcal{N}(\theta_y, \sigma_y^2/N_E) && (15)\\ \mu_{a,1} - \mu_{a,0} &\sim \mathcal{N}(\theta_a, \sigma_a^2/N_E) && (16)\\ \mu_{d,1} - \mu_{d,0} &\sim \mathcal{N}(\theta_d, \sigma_d^2/N_E) && (17)\\ \mu_{v,1} - \mu_{v,0} &\sim \mathcal{N}(\theta_v, \sigma_v^2/N_E) && (18)\end{aligned}$$

Hence we can follow the aforementioned approach to get the variance. Then we use equation (10) to get α_0 and $\alpha_1 = 1 - \alpha_0$. Finally we reuse the predefined penalty ζ_0 and ζ_1 to accept (or reject) the hypotheses.

2.2 Frequentist Approach

We adopted the simple t-test for all the features. To begin with, we tested the treatment effect of 'sumX' both in the control and the treatment group. Following the above notation, we have the following relations

$$\frac{\big((\overline{X}_{1,1} - \overline{X}_{1,0}) - (\overline{X}_{0,1} - \overline{X}_{0,0})\big) - (\mu_{x,1} - \mu_{x,0})}{\sqrt{S_{x,1}^2/N_T + S_{x,0}^2/N_C}} \qquad (19)$$

$$\sim t_{N_C+N_T-2} \qquad (20)$$

The numerator of equation (19) shares the same format as equation (5), and the $S_{x,1}^2$ and $S_{x,0}^2$ are the sample variances of $(X_{1,1,l} - X_{1,0,l})$ and $(X_{0,1,k} - X_{0,0,k})$ respectively. We ran the t-test and found that the 95% confidence interval is $[-0.02576, 0.01133]$. And the corresponding p-value is 0.4455. In fact we make the boxplot to show the experiment effect is close to 0 in Figure 1. Therefore we accept the null hypothesis, i.e. $\theta_x = 0$, and there is no significant difference in feature 'sumX' between pre- and post-experiment.

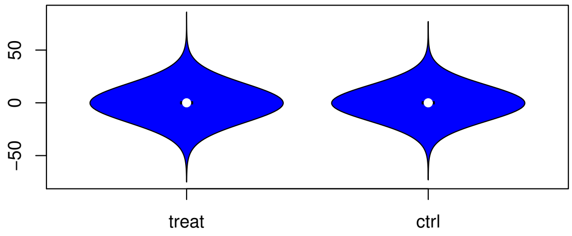

Figure 1: Comparison of experiment's effects on the control and the treatment groups on feature 'sumX'. The white dot in the middle is the mean that is close to zero

The same procedures were also applied to other features. We apply the **Benjamini-Hochberg procedure** to conduct test on multiple hypotheses [1] and present the results in the following section.

3 Experiment Results

3.1 Bayes Approach Result

Table 2 depicts the Bayes approach result. In the calculation we chose $\pi_0 = 0.9$ because we believe that many features won't change significantly after the expreiment. Also the marginal distribution standard deviation V was calculated according to the majority of the group samples. Here majority means we trim-off the lower 1% and upper 99% quantile samples treated as outliers. All features satisfy the condition such that $\alpha_0 > \frac{\zeta_0}{\zeta_1}\alpha_1$. Thus we accept the null hypotheses that there are no significant changes on those 5 features from pre- to post-experiment. Therefore we should not deploy the new changes.

However, if we tune the prior $\pi_0 = 0.55$, which means we still believe that there will be some change in features. The corresponding results are more or less similar, except we reject the null hypothesis for feature *sumY*, since $\alpha_0 < \frac{\zeta_0}{\zeta_1}\alpha_1$ according to the bolded number in Table 3. Different prior will yield different decisions according to our previous settings. These decisions also related to the loss function value, namely ζ_0 and ζ_1 in our case. In fact we plot out the box-plot (Figure 3 in the appendix) for the feature *sumY* and find that there is a heavy tail located in the positive part (above value 2000) for the treatment group. This may imply that there could be a treatment effect for feature *sumY*.

Table 2: Posterior probability results, $\pi_0 = 0.9$

feature	$\alpha_0 = P(H_0\|x)$	$\alpha_1 = P(H_1\|x)$	$\frac{\zeta_0}{\zeta_1}\alpha_1$
sumX	0.997731	0.002269	0.022686
sumY	0.972921	0.027079	0.270793
PV	0.987139	0.012861	0.128614
NumDays	0.999399	0.000601	0.006010
NumVisits	0.998853	0.001147	0.011466

Table 3: Posterior probability results, $\pi_0 = 0.55$

feature	$\alpha_0 = P(H_0\|x)$	$\alpha_1 = P(H_1\|x)$	$\frac{\zeta_0}{\zeta_1}\alpha_1$
sumX	0.983533	0.016467	0.164673
sumY	$\mathbf{0.82991}^{*}$	0.17009	$\mathbf{1.70091}^{*}$
sumPV	0.912458	0.087542	0.875417
NumDays	0.995591	0.004409	0.044087
NumVisits	0.991618	0.008382	0.083820

3.2 Frequentist Approach Result

Table 4 shows the results of standard t-test for the 5 features. Because of the high p-value, we cannot reject any hypothesis. We also use the Benjamini-Hochberg procedure to decide whether to accept or reject these multiple hypotheses. After sorting the p-values of the associated hypotheses, we present Table 5, where θ is the mean of difference between the treatment group and the control group of the experiment. For example, $\theta_x = \mathbb{E}(\mu_{x,1} - \mu_{x,0})$. In this experiment, given $\alpha = 0.05$, we find the maximum index k such that

$$p_{(k)} \leq \frac{k\alpha}{m \cdot c(m)} \tag{21}$$

where $m = 5$ is the total number of hypotheses, and $c(m) = 1$ due to the independent or positive regression dependent assumption. It turns out that all hypotheses are accepted, and the false discovery rate (FDR) is less than or equal to $\alpha = 0.05$.

Table 4: Regular t-test results

feature	p-value	95% CI
sumX	0.4455351	(-0.02576341, 0.01132676)
sumY	0.7381790	(-0.15781821, 0.11183053)
PV	0.7131157	(-0.11338964, 0.07756754)
numDays	0.6868866	(-0.00551990, 0.00837815)
numVisits	0.9683994	(-0.01954911, 0.02035569)

Table 5: Ordered hypotheses by p-value

feature	index	H_0	p-value
sumX	$H_{(1)}$	$\theta_x = 0$	0.4455351
numDays	$H_{(2)}$	$\theta_d = 0$	0.6868866
PV	$H_{(3)}$	$\theta_a = 0$	0.7131157
sumY	$H_{(4)}$	$\theta_y = 0$	0.7381790
numVisits	$H_{(5)}$	$\theta_v = 0$	0.9683994

Additionally, we would like to investigate the variance reduction technique, namely the **Control Variate**, to help improve the hypotheses testing [3]. Here we analyze the feature 'sumX' as an example. Following the previous notation, let $X_{1,0,l}$ and $X_{1,1,l}$ be the 'sumX' metric before the experiment and after the experiment. Let new sample $U_{1,l}$ and $U_{0,k}$ be

$$U_{1,l} = X_{1,1,l} - X_{1,0,l} \tag{22}$$

$$U_{0,k} = X_{0,1,k} - X_{0,0,k} \tag{23}$$

Now we introduce the new samples $V_{1,l}$ and $V_{0,k}$ such that

$$V_{1,l} = U_{1,l} - b(X_{1,0,l} - \mathbb{E}(X_{1,0,l})) \tag{24}$$

$$V_{0,k} = U_{0,k} - b(X_{0,0,k} - \mathbb{E}(X_{0,0,k})) \tag{25}$$

since we assume that the treatment effect for the treatment sample $U_{1,l}$ has some correlation with the pre-experiment sample $X_{1,0,l}$. Similarly we assume there is correlation between $U_{0,l}$ and $X_{0,0,l}$ as well. To express this in the variable notation, we drop the subscription l, k. The effect $\overline{U}_1 - \overline{U}_0$ is closely related to $\overline{V}_1 - \overline{V}_0$ as

$$\mathbb{E}(\overline{V}_1 - \overline{V}_0) = \tag{26}$$

$$\mathbb{E}\Big(\overline{U}_1 - b(\overline{X}_{1,0} - \mathbb{E}(X_{1,0})) - \overline{U}_0 - b(\overline{X}_{0,0} - \mathbb{E}(X_{0,0}))\Big) \tag{27}$$

$$= \mathbb{E}(\overline{U}_1 - \overline{U}_0) = \mathbb{E}\Big((\overline{X}_{1,1} - \overline{X}_{1,0}) - (\overline{X}_{0,1} - \overline{X}_{0,0})\Big) \tag{28}$$

Notice that we can simulate b by leveraging the following equation

$$b = \frac{\mathrm{Cov}(U_i, X_{i,0})}{\mathrm{Var}\,(X_{i,0})}, \tag{29}$$

where the b calculated from the treatment group when $i = 1$, and b obtained from the control group when $i = 0$. b can be different for the treatment and the control group. But in our implementation, we assume the correlation effects are similar. Hence we use Monte Carlo method to approximate the b value and use it consistently. By utilizing this attribute, we use the following test statistic:

$$Z = \frac{\overline{V}_1 - \overline{V}_0}{\sqrt{\mathrm{Var}(\overline{V}_1) + \mathrm{Var}(\overline{V}_0)}} \sim \mathcal{N}(0, 1) \tag{30}$$

In practice, the $\mathrm{Var}(\overline{V}_0)$ and $\mathrm{Var}(\overline{V}_1)$ are replaced by the sample variance. Thus the normal approximation is adopted to check the null hypothesis. We apply the same logic to other features and display the 95% confidence interval for both the non-control variate and control variate cases in Figure 2. The control variate approach has lower variance and yields narrower confidence intervals than the non-control variance case.

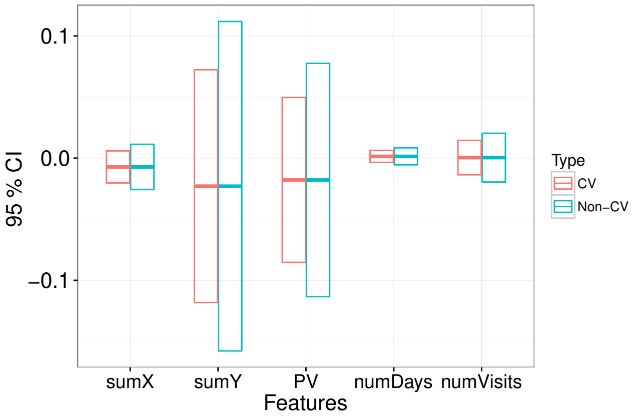

Figure 2: 95% CI of null hypotheses for multiple features in both Non-Control variate and Control variate cases

Again from the result in Table 6, we can not reject any null hypothesis because the 95% confidence interval covers 0.

Table 6: Control variate test result

feature	p-value	95% CI
sumX	0.4454621	(-0.020337646 0.005895250)
sumY	0.7382217	(-0.118260478 0.072305642)
PV	0.7124793	(-0.085504124 0.049582867)
numDays	0.6867551	(-0.003483039 0.006341825)
numVisits	0.9691224	(-0.013719994 0.014508427)

4 Conclusion & Discussion

We examine the experiment (treatment) effects of Bing search users based on typical features, e.g. total clicks (sumX), total browsing time (sumY), etc., through two approaches: Bayes rule and frequentist rule. Each approach has its own advantages and disadvantages. The Bayes approach compares both the posterior probability of null hypothesis, i.e. α_0, and the posterior probability of alternative hypothesis, i.e. α_1, given the associated loss ζ_0 and ζ_1. It attempts to minimize the expected loss for all the possible actions, in this case accept or reject the hypothesis. In our setting of the 0-constant loss function, the decision rule doesn't only depend on α_0, but also on α_1, ζ_1 and ζ_2. The subtlety argument is that when we calculate the posterior probability, the certain parameters have to be considered as known. In our implementation, the prior π_0 and the marginal density variance V^2 are treated as given. When we calculate the posterior probabilities for corresponding hypotheses, the decision could be varied by the different loss functions, which can be related to how people perceive the penalty of each decision.

The frequentist rule carries the attribute of easy-calculation. Essentially it computes the α_0 and the associated p-value. The reason supports this approach is based on *Law of Large Number* and *Central Limit Theorem*. It doesn't necessarily take into account the loss function. But it provides a very generic framework to test hypotheses. The concept of false discovery rate has been adopted to make the decision on multiple hypotheses much quicker by incorporating the Benjamini-Hochberg procedure. We also use the control variate technique to improve the variance of the test statistics, which yields a narrower confidence interval than the approach of the regular frequentist test.

More extensively, if the sample size is small, we prefer the Bayes rule approach to support our decisions. If the sample size is large, for instance, having over tens of thousands samples, then we adopt the hypothesis testing approach. There is an alternative way to detect the treatment effect as well, which is running a simple linear regression to check whether the coefficient, particularly β_3 in our case, equals to 0. The zero coefficient means no treatment effect. Then we can run the hypothesis testing for $H_0 : \beta_3 = 0$ to either accept or reject the null hypothesis. We will explore this approach in the future.

References

[1] Y. Benjamini and Y. Hochberg, *Controlling the false discovery rate: a practical and powerful approach to multiple testing*, Journal of the royal statistical society. Series B (Methodological), (1995), pp. 289–300.

[2] A. Deng, *Objective bayesian two sample hypothesis testing for online controlled experiments*, in Proceedings of the 24th International Conference on World Wide Web, ACM, 2015, pp. 923–928.

[3] A. Deng, Y. Xu, R. Kohavi, and T. Walker, *Improving the sensitivity of online controlled experiments by utilizing pre-experiment data*, in Proceedings of the sixth ACM international conference on Web search and data mining, ACM, 2013, pp. 123–132.

[4] A. Diamond and P. S. Goldman-Rakic, *Comparison of human infants and rhesus monkeys on piaget's ab task: Evidence for dependence on dorsolateral prefrontal cortex*, Experimental brain research, 74 (1989), pp. 24–40.

5 Appendix

5.1 Plots

We attach more plots to compare the treatment effect with the details.

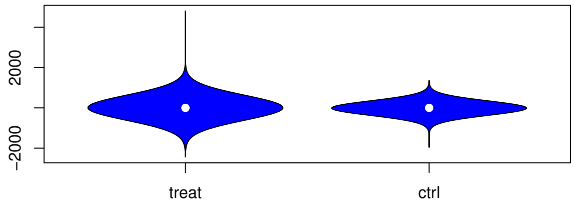

Figure 3: feature $sumY$

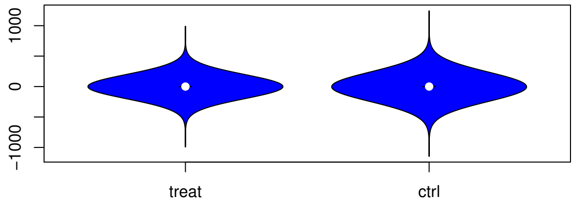

Figure 4: feature PV

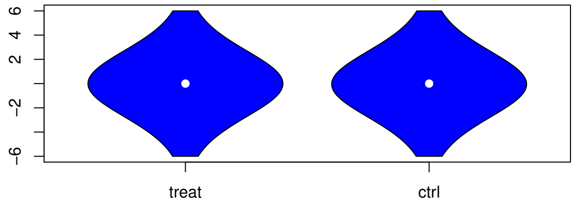

Figure 5: feature $numDays$

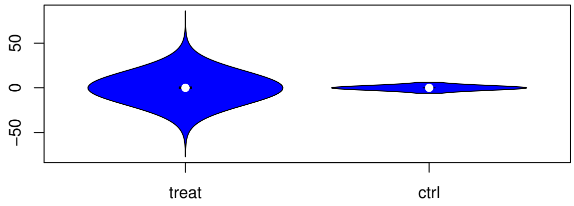

Figure 6: feature $numVisits$

5.2 Code

```
if (!require(data.table)) install.packages('data.table')
library(data.table)
library(ggplot2)

pwd = "../.." # set your working dir
# setwd("File location")
setwd(pwd)

load(paste(pwd,"data.RData", sep=""))

# dataset:
# each row is a record of a user's Bing usage before or after the experiment
# group is treatment assignment. 1: treatment, 0: control. Randomization is on userID
# IsPreExperiment indicates whether the page view is in AA period or AB period

rounding_err = 0.0000001

data_control_preEx = with(userLevel, userLevel[( Group > 0 - rounding_err & Group < 0 + rounding_err ) &
 (IsPreExperiment > 1 - rounding_err & IsPreExperiment < 1 + rounding_err ) , ]) # data(0, 0, n)
data_control_postEx = with(userLevel, userLevel[( Group > 0 - rounding_err & Group < 0 + rounding_err ) &
 (IsPreExperiment > 0 - rounding_err & IsPreExperiment < 0 + rounding_err ) , ]) # data(0, 1, n)
data_treat_preEx = with(userLevel, userLevel[( Group > 1 - rounding_err & Group < 1 + rounding_err  ) &
(IsPreExperiment > 1 - rounding_err & IsPreExperiment < 1 + rounding_err ) , ]) # data(1, 0, n)
data_treat_postEx = with(userLevel, userLevel[( Group > 1 - rounding_err & Group < 1 + rounding_err ) &
 (IsPreExperiment > 0 - rounding_err & IsPreExperiment < 1 + rounding_err ) , ]) # data(1, 1, n)

feature_sumX_ctrl_pre = data_control_preEx$sumX
feature_sumX_ctrl_post = data_control_postEx$sumX
feature_sumX_treat_pre = data_treat_preEx$sumX
feature_sumX_treat_post = data_treat_postEx$sumX


#' @description
calculateAlpha = function( treat, ctrl, pi_0, V, desc=NULL ){
  #' @since
  #'

  # fig = vioplot( treat, ctrl, names = c("treat", "ctrl"), col = "blue") #+ title(xlab=desc, cex.lab=1.3)
  sigma_C_sq = var(treat)
  N_C = length(treat)
```

```r
  sigma_T_sq = var(ctrl)
  N_T = length(ctrl)
  N_E = 1/(1/N_C + 1/N_T )
  sigma_sq = (sigma_C_sq / N_C + sigma_T_sq / N_T) * N_E

  X_1 = treat
  X_0 = ctrl
  m_X_over_f_X = sigma_sq/(V*sqrt(N_E)) * exp(-((mean(X_1) - mean(X_0) )^2) * (1/2) * ( 1/(V^2) - 1/(sigma_sq/N_E)  ) )

  alpha_0 = (1 + (1-pi_0)/pi_0 * m_X_over_f_X)^(-1)
  alpha_1 = 1 - alpha_0
  #' @note
  #' PNT = Probability of Null hypothesis is True
  #' PNF = Probability of Null hypothesis is False
  return(list(PNT=alpha_0, PNF=alpha_1 ))
}

#' @description This function takes out the lower and upper
#'              quantile to extract the middle distribution mass
trim_outlier = function(x, q_low, q_up) {
   new_x = x[ x < quantile(x, q_up)]
   new_x = new_x[new_x > quantile(x, q_low)]
   return(new_x)
}


# V_x = sd(c(feature_sumX_treat_post - feature_sumX_treat_pre, feature_sumX_ctrl_post- feature_sumX_ctrl_pre))

pi_0 = 0.55  # 1 - p, a prior belief

V_x_slim = sd(trim_outlier(feature_sumX_ctrl_post-feature_sumX_ctrl_pre, 0.01, 0.99 ))

res_X = calculateAlpha(treat = (feature_sumX_treat_post - feature_sumX_treat_pre),
                       ctrl = (feature_sumX_ctrl_post - feature_sumX_ctrl_pre ),
                       pi_0 = pi_0 ,
                       # V=sd(feature_sumX_ctrl_post- feature_sumX_ctrl_pre),
                       V=V_x_slim,
                       desc = "sumX" )

# V_y = sd(c( data_treat_postEx$sumY - data_treat_preEx$sumY  , data_control_postEx$sumY - data_control_preEx$sumY ))

V_y_slim = sd(trim_outlier(data_control_postEx$sumY - data_control_preEx$sumY, 0.01, 0.99 ))
```

```
res_Y = calculateAlpha(treat = (data_treat_postEx$sumY - data_treat_preEx$sumY),
                       ctrl = (data_control_postEx$sumY - data_control_preEx$sumY),
                       pi_0 = pi_0,
                       # V= sd(data_control_postEx$sumY - data_control_preEx$sumY),
                       V=V_y_slim,
                       desc = "sumY")


V_pv = sd(c(data_treat_postEx$PV - data_treat_preEx$PV  , data_control_postEx$PV - data_control_preEx$PV ))
V_pv_slim = sd(trim_outlier(data_control_postEx$PV - data_control_preEx$PV, 0.01, 0.99 ))
res_PV = calculateAlpha(treat = (data_treat_postEx$PV - data_treat_preEx$PV ),
                        ctrl = (data_control_postEx$PV - data_control_preEx$PV),
                        pi_0 = pi_0,
                        # V=sd(data_control_postEx$PV - data_control_preEx$PV),
                        V=V_pv_slim,
                        desc = "PV")

# V_numDays = sd(c(data_treat_postEx$numDays - data_treat_preEx$numDays  , data_control_postEx$numDays - data_control_preEx$numDays ))

V_numDays_slim = sd(trim_outlier(data_control_postEx$numDays - data_control_preEx$numDays, 0.01, 0.99 ))

res_numDays = calculateAlpha(treat = (data_treat_postEx$numDays - data_treat_preEx$numDays ) ,
                             ctrl = (data_control_postEx$numDays - data_control_preEx$numDays),
                             pi_0 = pi_0 ,
                             # V= sd(data_control_postEx$numDays - data_control_preEx$numDays)  ,
                             V = V_numDays_slim,
                             desc = "numDays")


# V_numVisit = sd(c(data_treat_postEx$numVisits - data_treat_preEx$numVisits , data_control_postEx$numVisits - data_control_preEx$numVisits ))

V_numVisit_slim = sd(trim_outlier(data_control_postEx$numVisits - data_control_preEx$numVisits, 0.01, 0.99 ))

res_numVisit = calculateAlpha(treat = (data_treat_postEx$numVisits - data_treat_preEx$numVisits ) ,
                              ctrl = (data_control_postEx$numDays - data_control_preEx$numDays ) ,
                              pi_0 = pi_0,
                              # V= sd(data_control_postEx$numVisits - data_control_preEx$numVisits) ,
                              V= V_numVisit_slim,
                              desc = "numVisits")

#’ @description  displaying k-th decimal
```

```
formatted = function(x, k){
  return(format(round(x, k), nsmall=k))
}

print("========== Bayes =============")
print(paste( "X:         alpha_0 =", formatted(res_X$PNT, 6),    "alpha_1 =", formatted( res_X$PNF, 6 )) )
print(paste( "Y:         alpha_0 =", formatted(res_Y$PNT, 6),    "alpha_1 =", formatted(res_Y$PNF, 6) ))
print(paste( "PV:        alpha_0 =", formatted(res_PV$PNT, 6),   "alpha_1 =", formatted(res_PV$PNF, 6)))
print(paste( "NumDays:   alpha_0 =", formatted(res_numDays$PNT, 6),  "alpha_1 =", formatted(res_numDays$PNF,6)))
print(paste( "NumVisits: alpha_0 =", formatted(res_numVisit$PNT, 6), "alpha_1 =", formatted(res_numVisit$PNF,6 )))

zeta_0 = 10 #  K_0 panelty for H_0 is false
zeta_1 = 1 # K_1 panelty for H_1 is false (\mu \neq 0 is false )

#’ @description  feature collection
feature_col = rbind(res_X, res_Y, res_PV, res_numDays, res_numVisit)

for ( i in 1:(dim(feature_col)[1]) ){
  cat_sign = " = "
  decision = ""
  res = feature_col[i, ]
  if (res$PNT > ( (zeta_0/zeta_1)* res$PNF) ){
    cat_sign = " > "
    decision = " accept H_0"
  }else if ( res$PNT < ((zeta_0/zeta_1)*res$PNF) ) {
    cat_sign = " < "
    decision = " reject H_0"
  }
  print(paste( formatted(res$PNT, 6), cat_sign, formatted( (zeta_0/zeta_1)*res$PNF, 6 ), ",", decision , sep="" ))
}

#’ @end of the Bayes Rule
#’ ====================================

#’ @description Frequentist Approach
#’ ====================================

feat_name = c("sumX", "sumY", "PV", "numDays", "numVisits")

res_col = NULL
for (i in 1:length(feat_name)) {
  X = (data_treat_postEx[[feat_name[i]]] - data_treat_preEx[[feat_name[i]]]) - ( data_control_postEx[[feat_name[i]]] - data_control_preEx[[feat_name[i
  res = t.test(X, mu=0)
  tmp = data.frame( f_name = feat_name[i], p_val = res$p.value, conf_int_low = res$conf.int[1], conf_int_up = res$conf.int[2], estimate = mean(X) )
  if (is.null(res_col)){
    res_col = tmp
  }else {
    res_col = rbind(res_col, tmp)
  }

}
print("========== Freqentist =============")
print(res_col)

#' @note start Benjamini-Hochberg procedure
print("========== Benjamini ===========")
ordered_pVal_DF = res_col[order(res_col$p_val),]

alph_multi = 0.05
for (j in length(feat_name):1){
  if (ordered_pVal_DF[j,2] > (j*alph_multi/length(feat_name) ) ){
    print( paste("accept H_(", j, ");", sep = ""  ) )

  }else if (ordered_pVal_DF[j,2] <= (j*alph_multi/length(feat_name) ) ) {
    print( paste(" reject H_(",j,");" , sep = ""  ))
    print( paste("reject all rest H(k), where k < ", j, ";", sep="" ))
    break
  }
}

print("========= Control Variate ============")

res_cv_col = NULL
for (i in 1:5){
  n_pre = length(data_control_preEx[[feat_name[i]]])
  n_post = length(data_control_postEx[[feat_name[i]]])
  N = min(n_pre, n_post)
  u_1 = data_treat_postEx[[feat_name[i]]] - data_treat_preEx[[feat_name[i]]]
  u_0 = data_control_postEx[[feat_name[i]]] - data_control_preEx[[feat_name[i]]]

  b1_sim = NULL
  b0_sim = NULL
  for (t in 1:10){
    trimmed_u_1 = sample(u_1, size = N/5)
    trimmed_x_1 = sample(data_treat_preEx[[feat_name[i]]], N/5)
    b1_temp = cov(trimmed_u_1, trimmed_x_1 )/ var(trimmed_x_1)
    b1_sim = c(b1_sim, b1_temp )

    trimmed_u_0 = sample(u_0, size = N/3)
    trimmed_x_0 = sample(data_control_preEx[[feat_name[i]]], N/3)
    b0_temp = cov(trimmed_u_0, trimmed_x_0 )/ var(trimmed_x_0)
    b0_sim = c(b0_sim, b0_temp )
  }
  # b1 = mean(b1_sim)
  b0 = mean(b0_sim)
  # b0 = b1
  b1 = b0
  # b

  v_1 = u_1 - b1*data_treat_preEx[[feat_name[i]]]
  v_0 = u_0 - b0*data_control_preEx[[feat_name[i]]]

  V = v_1-v_0
  # var(v1)/length(v_1) + var(v_0)/length(v_0)
  res_cv = t.test(V, mu=0)  # result of control variate
  cv1_sd = sd(V)/sqrt(2*length(V))
  cv1_mu = mean(V)

  tmp_cv = data.frame( f_name = feat_name[i],
   p_val = res_cv$p.value,
   conf_int_low = cv1_mu - qnorm(0.975)*cv1_sd ,
   conf_int_up = cv1_mu + qnorm(0.975)*cv1_sd , estimate=cv1_mu)
  if (is.null(res_cv_col)){
    res_cv_col = tmp_cv
  }else {
    res_cv_col = rbind(res_cv_col, tmp_cv)
  }

}
print(res_cv_col)

res_cv_col$Type = rep("CV", 5)
res_col$Type = rep("Non-CV", 5)
# res_col$estimate = rep(0, 5)
```

```
combine_res = rbind(res_col, res_cv_col)

# combine_res$estimate = rep(0 , 10) # expected value are 0
dodge <- position_dodge(width=0.55)
fig2 = ggplot(data=combine_res, aes(f_name, estimate, colour=Type )) +
  geom_crossbar(aes(ymin = conf_int_low, ymax = conf_int_up), width=0.53, position = dodge ) +
  ylab("95 % CI ") + xlab("Features") +
  theme_bw() +
  theme(axis.text.y=element_text(size=16),
        axis.text.x=element_text(size=14),
        strip.text.x =element_text(size=16),
        axis.title.y = element_text(size=16),
        axis.title.x = element_text(size=16))

print(fig2)
```