A/B testing

Mingwei Wu

```
click_data<-read.csv(url("https://assets.datacamp.com/production/repositories/2292/datasets/4407050e9b8
click_data$visit_date<-as.Date(click_data$visit_date)
head(click_data)
```

```
##   visit_date clicked_adopt_today
## 1 2017-01-01                   1
## 2 2017-01-02                   1
## 3 2017-01-03                   0
## 4 2017-01-04                   1
## 5 2017-01-05                   1
## 6 2017-01-06                   0
```

```
library("tidyverse")
```

```
## -- Attaching packages --------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library("lubridate")
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
library("ggplot2")
min(click_data$visit_date)
```

```
## [1] "2017-01-01"
```

```
max(click_data$visit_date) #check the date range in the database
```

```
## [1] "2017-12-31"
```

```
click_data%>%
  group_by(wday(visit_date))%>%
  summarize(conversion_rate=mean(clicked_adopt_today)) #count the conversion rate for day of visit date
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 7 x 2
##   `wday(visit_date)` conversion_rate
##                <dbl>           <dbl>
## 1                  1           0.3
## 2                  2           0.277
## 3                  3           0.271
## 4                  4           0.298
## 5                  5           0.271
## 6                  6           0.267
## 7                  7           0.256
```

```
click_data_sum<-click_data%>%
  group_by(week(visit_date))%>%
  summarise(conversion_rate=mean(clicked_adopt_today))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
head(click_data_sum) #count the conversion_rate for week of conversion rate
```

```
## # A tibble: 6 x 2
##   `week(visit_date)` conversion_rate
##                <dbl>           <dbl>
## 1                  1           0.229
## 2                  2           0.243
## 3                  3           0.171
## 4                  4           0.129
## 5                  5           0.157
## 6                  6           0.186
```

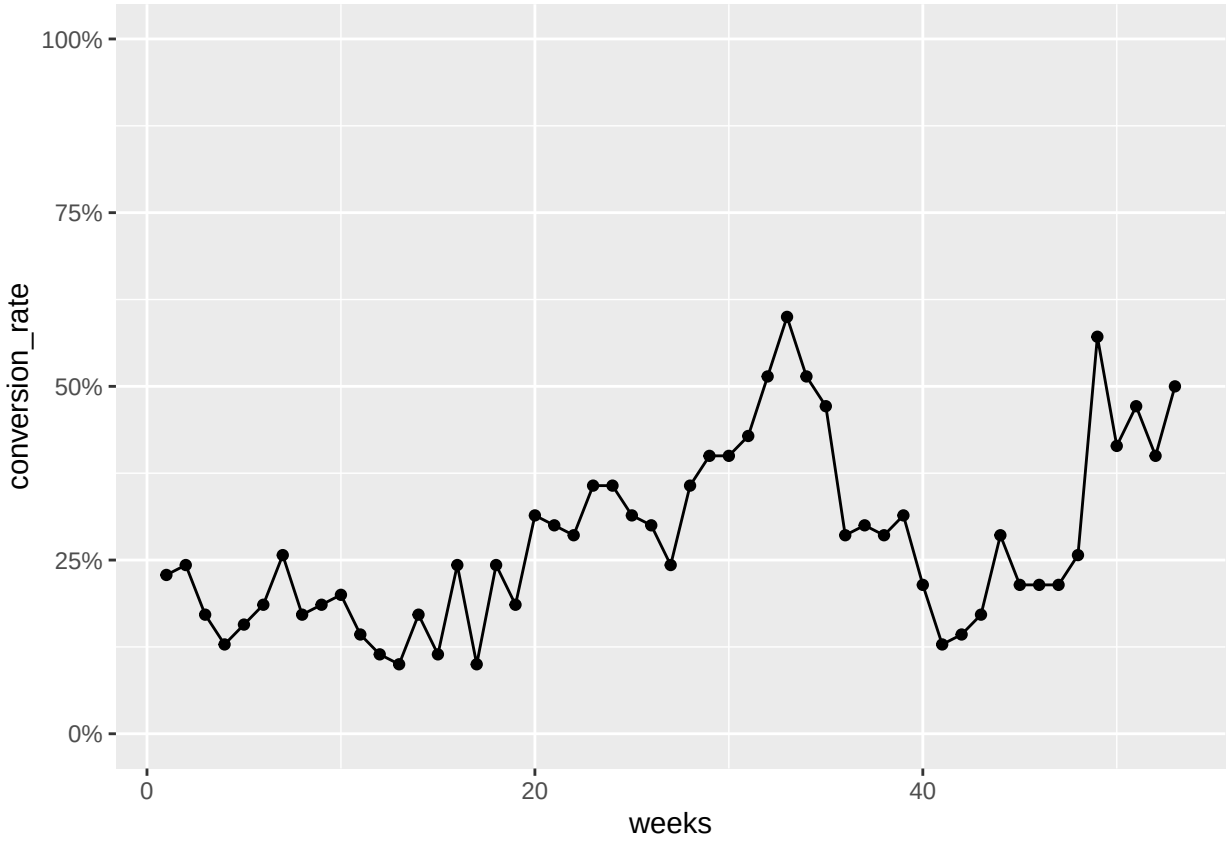
```
library("scales")
```

```
##
## Attaching package: 'scales'
```

```
## The following object is masked from 'package:purrr':
##
##     discard
```

```
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
click_data_sum%>%
  ggplot(aes(x=`week(visit_date)`,y=conversion_rate))+geom_point()+geom_line()+ scale_y_continuous(limi
```




```
library("powerMediation")
help("SSizeLogisticBin")
total_sample_size<- SSizeLogisticBin(p1= 0.54,
                                     p2= 0.64,
                                     B=0.5,
                                     alpha=0.05,
                                     power=0.8)
total_sample_size # Use the power-test to provide the sample size
```

```
## [1] 758
```

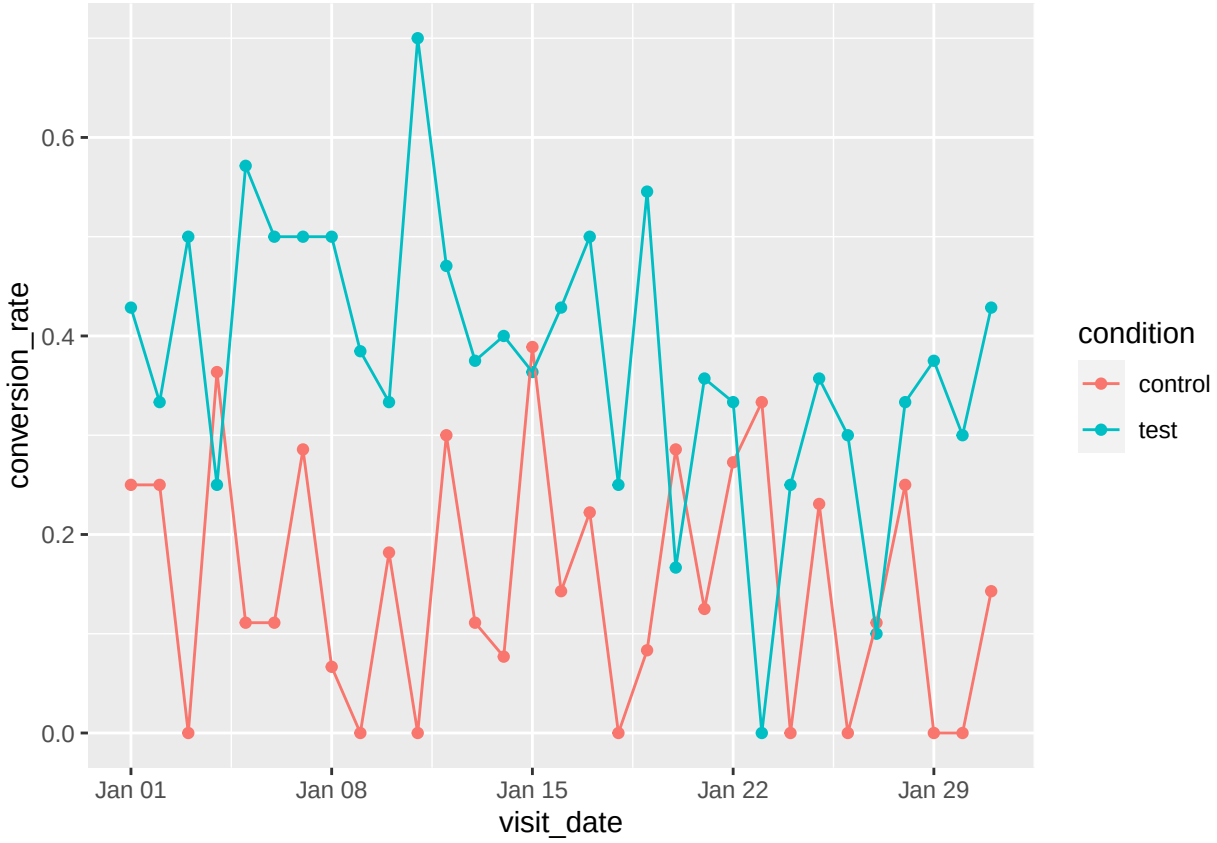
```
experiment_data<-read.csv(url("https://assets.datacamp.com/production/repositories/2292/datasets/52b52c
experiment_data$visit_date<-as.Date(experiment_data$visit_date)
head(experiment_data)# set up the A/B testing, control group and test group
```

```
##   visit_date condition clicked_adopt_today
## 1 2018-01-01   control                   0
## 2 2018-01-01   control                   1
## 3 2018-01-01   control                   0
## 4 2018-01-01   control                   0
## 5 2018-01-01      test                   0
## 6 2018-01-01      test                   0
```

```
experiment_data_sum<-experiment_data%>%
  group_by (visit_date , condition)%>%
  summarise(conversion_rate=mean(clicked_adopt_today))
```

```
## `summarise()` regrouping output by 'visit_date' (override with `.groups` argument)
```

```
experiment_data_sum%>%
  ggplot(aes(x=visit_date,
             y=conversion_rate,
             color= condition,group=condition))+geom_point()+geom_line() #plot between the control and
```




```
library(broom)
glm(clicked_adopt_today~condition,
    family = "binomial",
    data=experiment_data)%>%
  tidy() # doing the logistic regression for using condition to predict the cliked rate today.
```

```
## # A tibble: 2 x 5
##   term          estimate std.error statistic  p.value
##   <chr>            <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)      -1.61     0.156    -10.3  8.28e-25
## 2 conditiontest     1.14     0.197      5.77 7.73e- 9
```

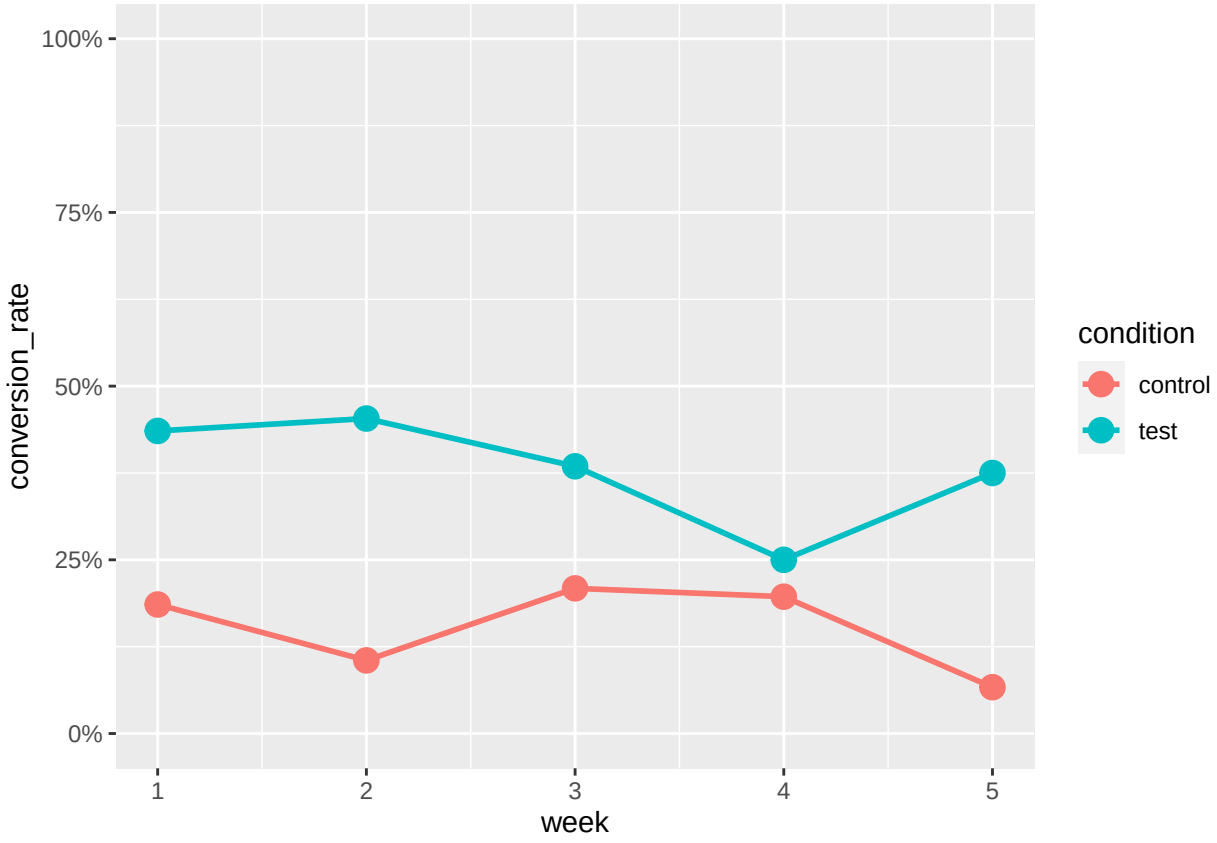
```
total_sample_size <- SSizeLogisticBin(p1 =0.39,
                                      p2=0.59,
                                      B =0.5,
                                      alpha=0.05,
                                      power=0.8)
total_sample_size #run logistic regression power analysis for sample size into our set up data
```

```
## [1] 194
```

```
month_click<-experiment_data%>%
  mutate(week = week(visit_date))%>%
  group_by(week,condition)%>%
  summarise(conversion_rate=mean(clicked_adopt_today))
```

```
## `summarise()` regrouping output by 'week' (override with `.groups` argument)
```

```
month_click%>%
  ggplot(aes(week,conversion_rate, color= condition, group= condition))+geom_point(size=4)+geom_line(lw
```




```
visit_website2018<-read.csv(url("https://assets.datacamp.com/production/repositories/2292/datasets/b502
head(visit_website2018)
```

```
##   visit_date condition time_spent_homepage_sec clicked_article
```

```
## 1 2018-04-01      tips               49.01161              1
## 2 2018-04-01      tips               48.86452              1
## 3 2018-04-01      tips               49.07467              1
## 4 2018-04-01      tips               49.26011              0
## 5 2018-04-01      tips               50.37190              0
## 6 2018-04-01      tips               49.08458              1
##   clicked_like clicked_share
## 1            0             1
## 2            0             0
## 3            0             0
## 4            1             0
## 5            1             0
## 6            0             0
```

```
library(broom)
ab_experiment<-glm(clicked_like~condition, family="binomial",data=visit_website2018)%>%
  tidy()   #runing logistic regression
ab_experiment
```

```
## # A tibble: 2 x 5
##   term           estimate std.error statistic   p.value
##   <chr>             <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)      -1.61     0.0219     -73.5 0.
## 2 conditiontools   -0.989    0.0390     -25.4 4.13e-142
```

```
ab_experiment_result<- t.test(time_spent_homepage_sec~condition,data=visit_website2018)  #run t-test
ab_experiment_result
```

```
##
##  Welch Two Sample t-test
##
## data:  time_spent_homepage_sec by condition
## t = 0.36288, df = 29997, p-value = 0.7167
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.01850573  0.02691480
## sample estimates:
##  mean in group tips mean in group tools
##            49.99909            49.99489
```

```
##sequential analysis
library(gsDesign)
# run sequential analysis
seq<-gsDesign(k=3,
              test.type = 1,
              alpha= 0.05,
              beta= 0.2,
              sfu="Pocock")  # k is look for how many times
seq
```

```
## One-sided group sequential design with
## 80 % power and 5 % Type I Error.
```

```
##            Sample
##             Size
##   Analysis Ratio*  Z   Nominal p  Spend
##          1  0.394 1.99    0.0232 0.0232
##          2  0.789 1.99    0.0232 0.0155
##          3  1.183 1.99    0.0232 0.0113
##      Total                       0.0500
##
## ++ alpha spending:
##  Pocock boundary.
## * Sample size ratio compared to fixed design with no interim
##
## Boundary crossing probabilities and expected sample size
## assume any cross stops the trial
##
## Upper boundary (power or Type I Error)
##           Analysis
##    Theta      1      2      3 Total   E{N}
##   0.0000 0.0232 0.0155 0.0113  0.05 1.1591
##   2.4865 0.3334 0.2875 0.1791  0.80 0.8070
```

```
# FILL in maxium number of points and compute points  per group and find stopping
max_n<-1500
max_per<-max_n /2
stopping_point<- max_per *seq$timing
stopping_point
```

```
## [1] 250 500 750
```