AB Test - Click Rate

Yingying Xu

Nov 2018

load package and data

```
library(sqldf)
```

```
## Loading required package: gsubfn
```

```
## Loading required package: proto
```

```
## Loading required package: RSQLite
```

```
data <- read.csv("abtest_example_ctr.csv")
summary(data)
```

```
##      userid      country         groups         deviceid
##  Min.   : 1000   CA:4571   control  :11460   Min.   : 5000
##  1st Qu.: 3256   CN:4576   treatment:11500   1st Qu.: 8758
##  Median : 5450   GB:4631                     Median :12538
##  Mean   : 5485   US:9182                     Mean   :12566
##  3rd Qu.: 7717                               3rd Qu.:16409
##  Max.   :10000                               Max.   :20000
##  NA's   :275
##      device        sellerid         itemid           date
##  Android:7003   Min.   :100.0   Min.   :1000   2017-05-16: 1721
##  Ios    :4583   1st Qu.:203.0   1st Qu.:1508   2017-05-15: 1701
##  Other  :4717   Median :304.0   Median :1994   2017-05-13: 1674
##  Web    :6657   Mean   :302.2   Mean   :1998   2017-05-09: 1657
##                 3rd Qu.:402.0   3rd Qu.:2497   2017-05-11: 1650
##                 Max.   :500.0   Max.   :3000   2017-05-17: 1650
##                                                (Other)   :12907
##      views            clicks         revenue
##  Min.   : 0.000   Min.   :0.000   Min.   :   0.00
##  1st Qu.: 4.000   1st Qu.:0.000   1st Qu.:   0.00
##  Median : 6.000   Median :1.000   Median :   0.00
##  Mean   : 5.796   Mean   :1.175   Mean   :  11.97
##  3rd Qu.: 7.000   3rd Qu.:2.000   3rd Qu.:   0.00
##  Max.   :20.000   Max.   :8.000   Max.   :1024.12
##
```

Note there are NA's for userid. Check the percentage of NA.

```
sum(is.na(data$userid))/nrow(data)
```

```
## [1] 0.01197735
```

Check for mixed assignment. There are 44 mixed assigned users

```
sqldf("select count(1) from (select userid, count(distinct(groups))
      from data group by userid having count(distinct(groups)) >1) as a")
```

```
##   count(1)
## 1       44
```

Check if multiple device per user, multiple user per device. There are 176 multiple devices per user, and 136 multiple users per device

```
sqldf("select count(1) from (select userid, count(distinct(deviceid))
      from data group by userid having count(distinct(deviceid))>1) as a")
```

```
##   count(1)
## 1      176
```

```
sqldf("select count(1) from (select deviceid, count(distinct(userid))
      from data group by deviceid having count(distinct(userid))>1) as a")
```

```
##   count(1)
## 1      136
```

Check NA/mixed/multiple device is random. create dummy, if any problems 1, else 0.

```
Pb_miss=1*is.na(data$userid)

userid_mix=as.numeric(sqldf("select userid from (select userid, count(distinct(groups))
                  from data group by userid having count(distinct(groups)) >1) as a")[[1]])
fun_Pb_mix=function(x){x %in%userid_mix}
Pb_mix=1*sapply(data$userid,FUN=fun_Pb_mix)

userid_mulD=as.numeric(sqldf("select userid from (select userid, count(distinct(deviceid))
                  from data group by userid having count(distinct(deviceid))>1) as a")[[1]])
fun_Pb_mulD=function(x){x %in%userid_mulD}
Pb_mulD=1*sapply(data$userid,FUN=fun_Pb_mulD)

deviceid_mulU=as.numeric(sqldf("select deviceid from (select deviceid, count(distinct(userid))
                  from data group by deviceid having count(distinct(userid)) >1) as a")[[1]])
fun_Pb_mulU=function(x){x %in%deviceid_mulU}
Pb_mulU=1*sapply(data$deviceid,FUN=fun_Pb_mulU)
```

For simplicity, I create a combined column 1/0 if any problems. In real projects, one may want to do this separately for each problem cuz they may have different causes.

```
data$pb_all = dd = apply(matrix(cbind(Pb_miss, Pb_mix, Pb_mulD, Pb_mulU), nrow = nrow(data)), 1, max)
```

Run model of dummy with other covariates, see if any covariates have strong correlation with having problematic assignment. Let's start with a simple model as there are not many covariates

```
pb_mod = glm(pb_all ~ country + groups + device + date + views + clicks + revenue, data, family = 'binom
summary(pb_mod)
```

```
##
## Call:
## glm(formula = pb_all ~ country + groups + device + date + views +
##     clicks + revenue, family = "binomial", data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.4988  -0.4186  -0.4028  -0.3846   2.3781
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -2.4432082  0.1297353 -18.832  < 2e-16 ***
## countryCN       -0.1284472  0.0764852  -1.679  0.09308 .
## countryGB       -0.1596440  0.0767825  -2.079  0.03760 *
## countryUS       -0.0976778  0.0652612  -1.497  0.13447
## groupstreatment -0.0280460  0.0488903  -0.574  0.56620
## deviceIos        0.1319837  0.0709948   1.859  0.06302 .
## deviceOther      0.1807197  0.0695398   2.599  0.00936 **
## deviceWeb        0.1128709  0.0645633   1.748  0.08043 .
## date2017-05-09  -0.0935126  0.1344048  -0.696  0.48658
## date2017-05-10  -0.0273475  0.1329418  -0.206  0.83702
## date2017-05-11   0.0285814  0.1313919   0.218  0.82780
## date2017-05-12   0.0908462  0.1295311   0.701  0.48309
## date2017-05-13   0.0347709  0.1302565   0.267  0.78951
## date2017-05-14   0.2118552  0.1274488   1.662  0.09646 .
## date2017-05-15  -0.0975244  0.1338009  -0.729  0.46608
## date2017-05-16  -0.0063666  0.1307095  -0.049  0.96115
## date2017-05-17   0.0381380  0.1305097   0.292  0.77012
## date2017-05-18  -0.0242176  0.1333616  -0.182  0.85590
## date2017-05-19   0.0491536  0.1311470   0.375  0.70781
## date2017-05-20   0.0376924  0.1310585   0.288  0.77365
## date2017-05-21  -0.0339439  0.1339728  -0.253  0.79999
## views            0.0036970  0.0113456   0.326  0.74454
## clicks          -0.0324605  0.0265073  -1.225  0.22073
## revenue          0.0002997  0.0006569   0.456  0.64823
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 12734  on 22959  degrees of freedom
## Residual deviance: 12709  on 22936  degrees of freedom
## AIC: 12757
##
## Number of Fisher Scoring iterations: 5
```

Some country, device have significant result, need to deep dive, if any bug exists, consider checking by type of problem (other device type have more deviced IDs? more likely to be missing?)

```
aggregate(Pb_miss, by = list(data$device), FUN = mean)
```

```
##   Group.1           x
## 1 Android 0.013708411
## 2     Ios 0.008946105
## 3   Other 0.009751961
## 4     Web 0.013820039
```

```
aggregate(Pb_mix, by = list(data$device), FUN = mean)
```

```
##   Group.1          x
## 1 Android 0.01870627
## 2     Ios 0.01352826
## 3   Other 0.01738393
## 4     Web 0.02268289
```

```
aggregate(Pb_mulD, by = list(data$device), FUN = mean)
```

```
##   Group.1          x
## 1 Android 0.05269170
## 2     Ios 0.05913157
## 3   Other 0.05554378
## 4     Web 0.05858495
```

```
aggregate(Pb_mulU, by = list(data$device), FUN = mean)
```

```
##   Group.1          x
## 1 Android 0.02970156
## 2     Ios 0.03753000
## 3   Other 0.04388382
## 4     Web 0.02974313
```

Other device has one ID mapping multiple users. Is this expected? Talk to Engineers, is this Logging problem?

blox of key metrics, views, clicks, ctr, problem

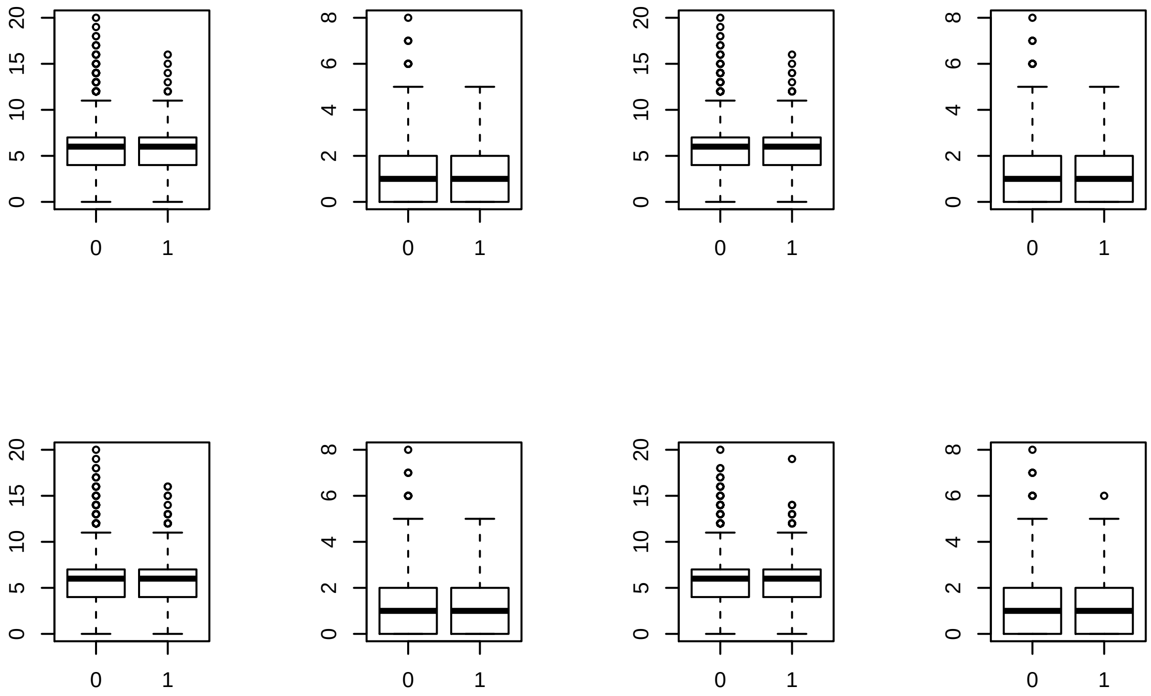
```
par(mfrow = c(2,4))

boxplot(data$views~Pb_miss)
boxplot(data$clicks~Pb_miss)

boxplot(data$views~Pb_mix)
boxplot(data$clicks~Pb_mix)

boxplot(data$views~Pb_mulD)
boxplot(data$clicks~Pb_mulD)

boxplot(data$views~Pb_mulU)
boxplot(data$clicks~Pb_mulU)
```



For simplicity in this work, I throw away the problematic assignments. In reality, checks carefully.

```
data=data[data$pb_all == 0,]
```

sanity check: check before experiment, metrics are comparable, no sig diff between test/control

Day1-day3 data was before experiment start

```
data$date=as.Date(data$date)
data_before<-data[data$date<(min(data$date)+3),]
data_start<-data[data$date>=(min(data$date)+3),]
```

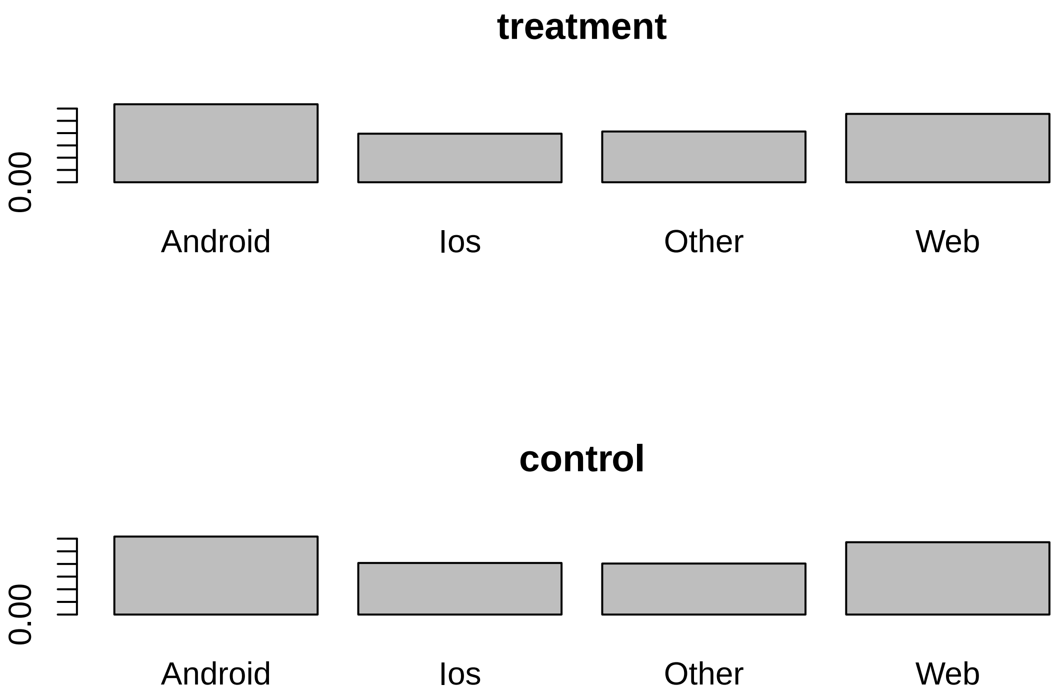
Compare aggregated CTR between test/control before experiment start; also compare view and click.

```
x1=sum(data_before$clicks[data_before$groups=='treatment'])
x2=sum(data_before$clicks[data_before$groups=='control'])
n1=sum(data_before$views[data_before$groups=='treatment'])
n2=sum(data_before$views[data_before$groups=='control'])
prop.test(x=c(x1,x2),n=c(n1,n2),alternative = 'two.sided')
```

```
##
##  2-sample test for equality of proportions with continuity
##  correction
##
## data:  c(x1, x2) out of c(n1, n2)
## X-squared = 0.0027542, df = 1, p-value = 0.9581
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.009858009  0.010568392
## sample estimates:
##    prop 1    prop 2
## 0.1985905 0.1982353
```

Compare other covariates comparable, take device as example.

```
par(mfrow = c(2,1))
barplot(prop.table(table(data_before[data_before$groups == 'treatment','device'])), main = 'treatment')
barplot(prop.table(table(data_before[data_before$groups == 'control','device'])), main = 'control')
```



Hypothesis Testing

Run test, not significant

```
x1=sum(data_start$clicks[data_start$groups=='treatment'])
x2=sum(data_start$clicks[data_start$groups=='control'])
n1=sum(data_start$views[data_start$groups=='treatment'])
n2=sum(data_start$views[data_start$groups=='control'])
prop.test(x=c(x1,x2),n=c(n1,n2),alternative = 'two.sided')
```

```
##
##  2-sample test for equality of proportions with continuity
##  correction
##
## data:  c(x1, x2) out of c(n1, n2)
## X-squared = 3.0668, df = 1, p-value = 0.07991
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.0005354551  0.0095636677
## sample estimates:
##    prop 1    prop 2
## 0.2063633 0.2018492
```

By subgroup, write a function, apply, which signficant

```
ztest_by_subgroup<-function(data_start, bycol, val)
{
  data_use=data_start[data_start[bycol]==val,]
  x1=sum(data_use$clicks[data_use$groups=='treatment'])
  x2=sum(data_use$clicks[data_use$groups=='control'])
  n1=sum(data_use$views[data_use$groups=='treatment'])
  n2=sum(data_use$views[data_use$groups=='control'])
  return(prop.test(x=c(x1,x2),n=c(n1,n2),alternative = 'two.sided'))
}

test_bydevice = data.frame(matrix(nrow = 0, ncol = 6,
                          dimnames = list(NULL,
                                          c('device','p.value','ctr_treatment',
                                            'ctr_control', 'ci.low','ci.high'))))
for (i in 1:length(unique(data$device))){
  device = as.character(unique(data$device)[i])
  test = ztest_by_subgroup(data_start, 'device', device)
  # you can check available statistics using names(test)
  testresult = data.frame('device' = device,
                          'p.value' = test$p.value,
                          'ctr_treatment' = test$estimate[1],
                          'ctr_control' = test$estimate[2],
                          'ci.low' = test$conf.int[1],
                          'ci.high' = test$conf.int[2])
  test_bydevice = rbind(test_bydevice,testresult)
}
test_bydevice
```

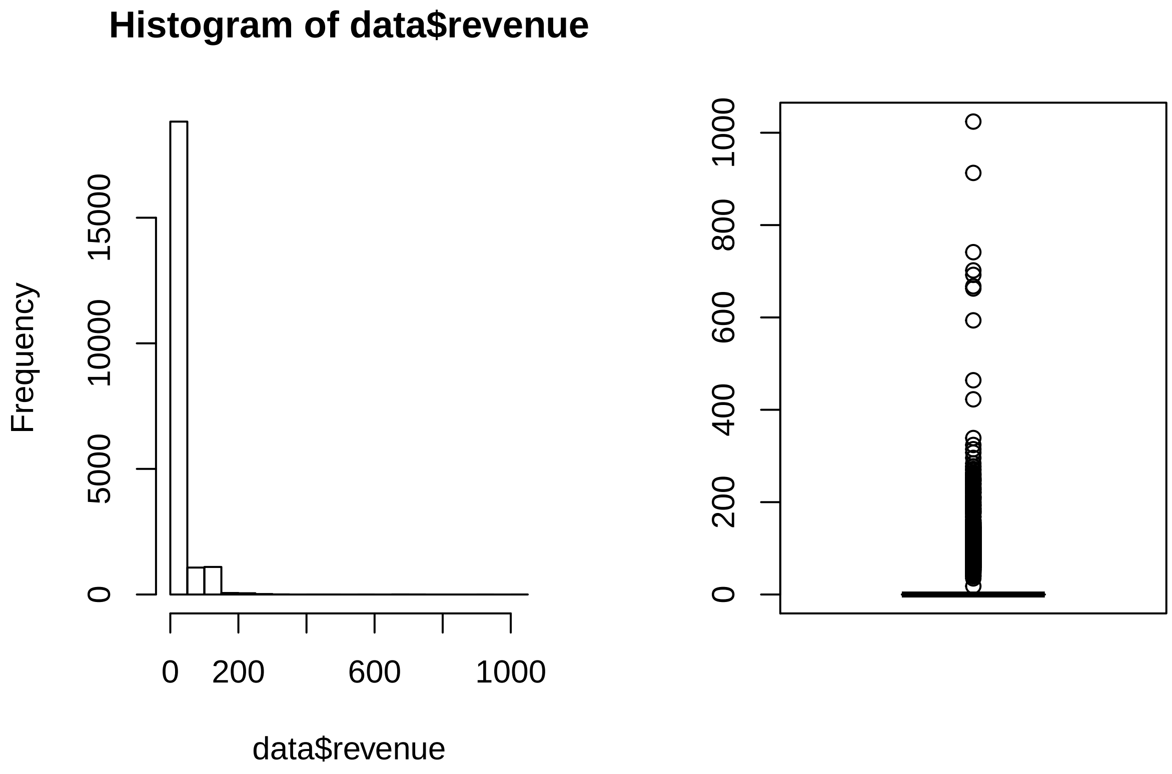
```
##          device      p.value ctr_treatment ctr_control       ci.low
## prop 1      Ios 0.0003394839     0.2239205   0.2027841  0.009515641
## prop 11 Android 0.3920080538     0.2012639   0.2053235 -0.013270662
## prop 12   Other 0.5348455174     0.2018839   0.1982983 -0.007523771
## prop 13     Web 0.4805146046     0.2036338   0.2002436 -0.005907063
##             ci.high
## prop 1  0.032757146
## prop 11 0.005151633
## prop 12 0.014694897
## prop 13 0.012687465
```

Ios significant, figure out why only works for Ios. Discuss with Eng & PM

Revenue

```
par(mfrow = c(1,2))
```

```r
hist(data$revenue)
boxplot(data$revenue)
```



The revenue is highly skewed.

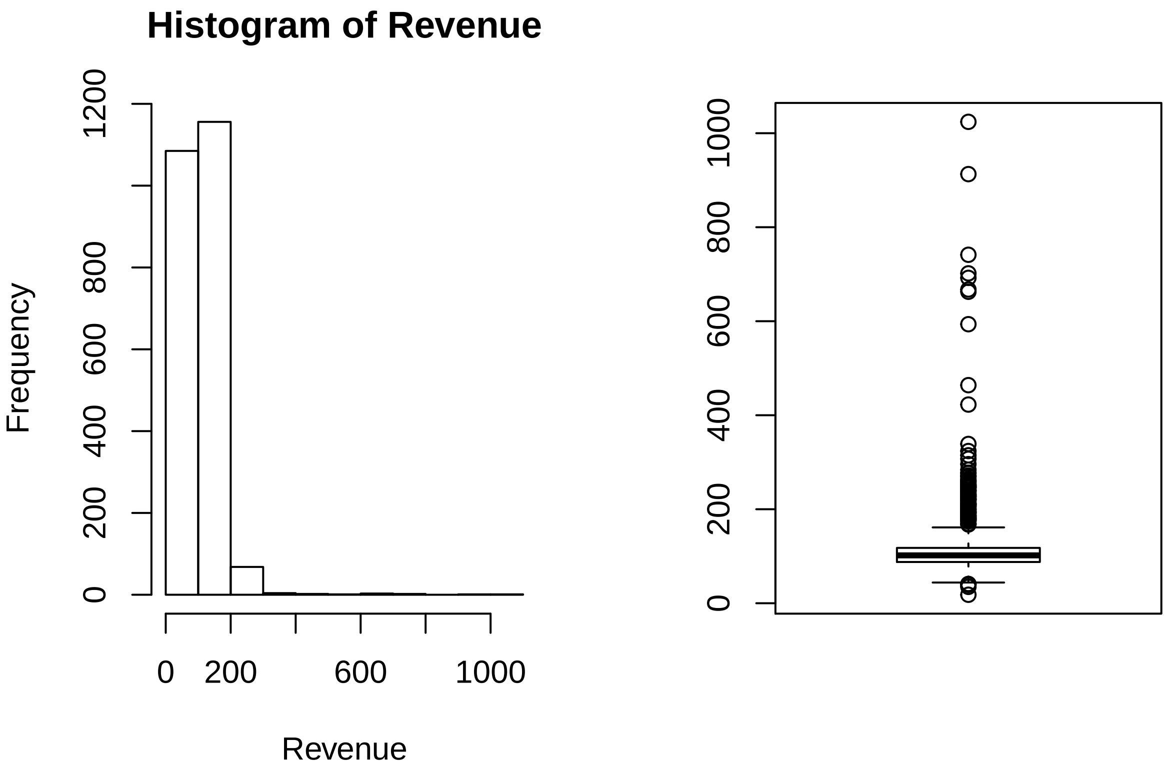
```r
sum(data$revenue == 0)/nrow(data)
```

```
## [1] 0.8900979
```

I drop those with revenue=0, only consider revenue with revenue>0

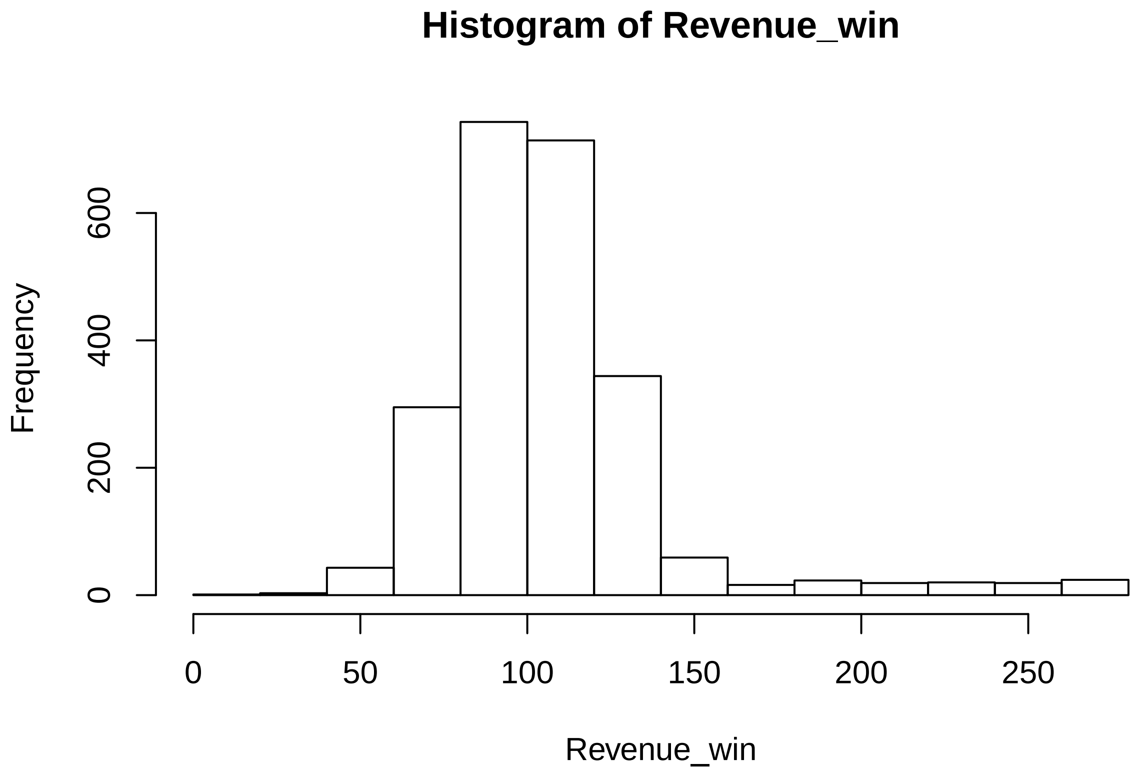
```r
Revenue<-data$revenue[data$revenue>0]

par(mfrow = c(1,2))
hist(Revenue)
boxplot(Revenue)
```



Winsorization, Capping

```
bound=quantile(Revenue,0.99)
Revenue_win<-Revenue
Revenue_win[Revenue>bound]=bound
hist(Revenue_win)
```



```
# or do it with built-in function
#library(robustHD)
#Winsorize(data$revenue, minval = NULL, maxval = NULL, probs = c(0.05, 0.95),  na.rm = FALSE)
```

Compare normal(CLT) & bootstrap distribution for estimator

```
#CLT
E_mean=mean(Revenue_win)
E_var=var(Revenue_win)/length(Revenue_win)

#bootstrap
True=mean(Revenue_win)
btsample = rep(0, 1000)
for (i in 1:1000){
  sample = Revenue_win[sample(length(Revenue_win), length(Revenue_win), replace =TRUE)]
  btsample[i] = mean(sample)
}
var_bt = var(btsample)

E_var
```

```
## [1] 0.4937082
```

```
var_bt
```

```
## [1] 0.4958332
```

What if the statistics of interest is 75% percentile of revenue if has spending?

```
mean = quantile(Revenue_win, 0.75)
btsample = rep(0, 1000)
for (i in 1:1000){
  sample = Revenue_win[sample(length(Revenue_win), length(Revenue_win), replace =TRUE)]
  btsample[i] = quantile(sample, 0.75)
}
var_bt = var(btsample)
var_bt
```

```
## [1] 0.7119308
```

Result Analysis

Regression Adjustment & diff-in-diff analysis

```
bound=quantile(Revenue,0.999)
data_before$revenue_win = ifelse(data_before$revenue>bound, bound, data_before$revenue)
data_start$revenue_win = ifelse(data_start$revenue>bound, bound, data_start$revenue)
```

regular t.test

```
x1=data_start$revenue_win[data_start$groups=='treatment']
x2=data_start$revenue_win[data_start$groups=='control']
t.test(x = x1, y = x2,alternative = 'two.sided')
```

```
##
##  Welch Two Sample t-test
##
## data:  x1 and x2
## t = -1.3746, df = 16606, p-value = 0.1693
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.957032  0.343613
## sample estimates:
## mean of x mean of y
##  11.93258  12.73929
```

There might be bias prior to experiment start

```
daily_rev_post = sqldf("select userid, country, device, groups, sum(revenue_win)/11 as rev_post
                       from data_start group by 1,2,3,4")
daily_rev_pre = sqldf("select userid, country, device, groups, date, sum(revenue_win)/3 as rev_pre
                      from data_before group by 1,2,3,4")
daily_rev = sqldf('select a.*, coalesce(rev_pre,0) as rev_pre from daily_rev_post a left outer join dai
```

diff in diff t test

```
x1= with(daily_rev[daily_rev$groups=='treatment',], rev_post - rev_pre)
x2= with(daily_rev[daily_rev$groups=='control',], rev_post - rev_pre)
t.test(x = x1, y = x2,alternative = 'two.sided')
```

```
##
##  Welch Two Sample t-test
##
## data:  x1 and x2
## t = -2.0599, df = 6728.5, p-value = 0.03945
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.08915080 -0.02697559
## sample estimates:
## mean of x mean of y
## 0.3476854 0.9057486
```

regression adjustment pre diff

```
rev_mod = lm(rev_post ~ groups + country + device + rev_pre, data = daily_rev)
summary(rev_mod)
```

```
##
## Call:
## lm(formula = rev_post ~ groups + country + device + rev_pre,
##     data = daily_rev)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.472 -2.800 -2.620 -2.285 67.837
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      2.938477   0.194347  15.120   <2e-16 ***
## groupstreatment -0.201092   0.133591  -1.505   0.1323
## countryCN       -0.036214   0.211067  -0.172   0.8638
## countryGB        0.104699   0.211098   0.496   0.6199
## countryUS       -0.117565   0.183826  -0.640   0.5225
## deviceIos       -0.041671   0.192653  -0.216   0.8288
## deviceOther     -0.041821   0.191202  -0.219   0.8269
## deviceWeb       -0.334607   0.172346  -1.941   0.0522 .
## rev_pre          0.007143   0.006759   1.057   0.2907
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.547 on 6894 degrees of freedom
## Multiple R-squared:  0.001357,   Adjusted R-squared:  0.0001979
## F-statistic: 1.171 on 8 and 6894 DF,  p-value: 0.3126
```

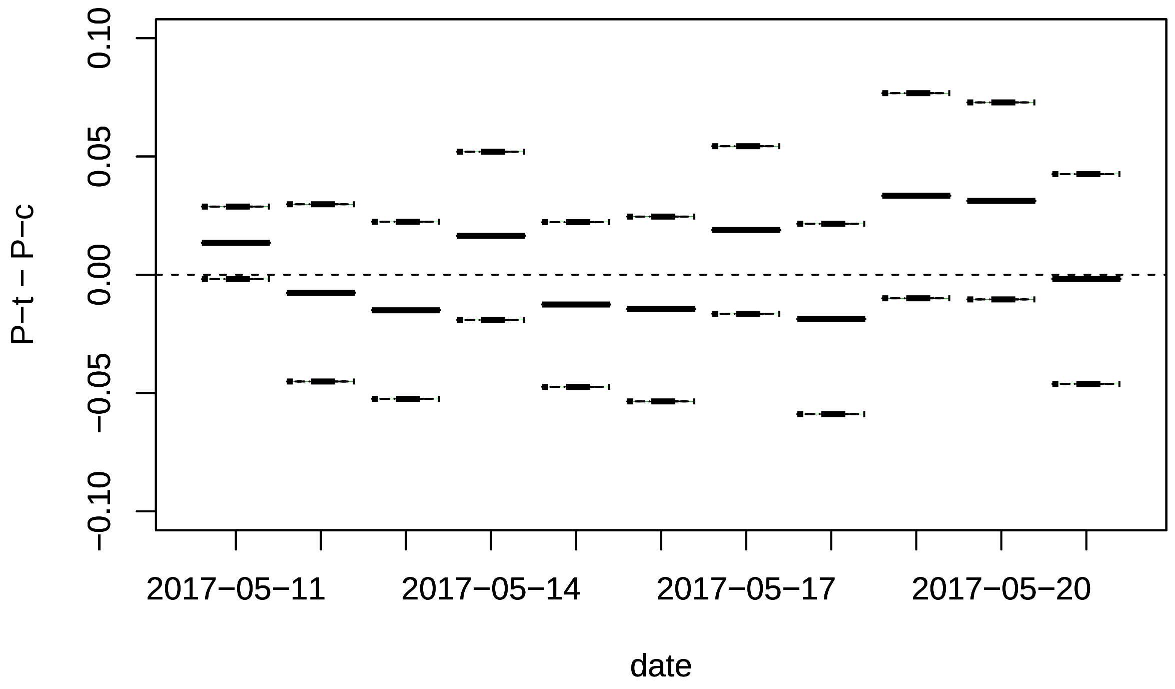
cohort analysis

change, over time change, 14 dates, 4th day start, line chart with CI by date. look at users enrolled on day 4.

```
d4 = data_start[data_start$userid %in% data_start[data_start$date == "2017-05-11", 'userid'],]
test_bydate = data.frame(matrix(nrow = 0, ncol = 6,
                                dimnames = list(NULL,
                                                c('date','p.value','ctr_treatment',
                                                    'ctr_control', 'ci.low','ci.high'))))

for (i in 1:(length(unique(data$date))-3)){
  date = as.character(sort(unique(data$date))[i+3])
  test = ztest_by_subgroup(d4, 'date', date)
  # you can check available statistics using names(test)
  testresult = data.frame('date' = date,
                          'p.value' = test$p.value,
                          'ctr_treatment' = test$estimate[1],
                          'ctr_control' = test$estimate[2],
                          'ci.low' = test$conf.int[1],
                          'ci.high' = test$conf.int[2])
  test_bydate = rbind(test_bydate,testresult)
}

par(mfrow = c(1,1))
plot(ctr_treatment - ctr_control ~ date, data = test_bydate, ylim = c(-0.1, 0.1), ylab = 'P-t - P-c')
plot(ci.low ~ date, data = test_bydate, lty = 4, col = 3, add = TRUE, ylab = '')
plot(ci.high ~ date, data = test_bydate, lty = 4, col = 3, add = TRUE, ylab = '')
abline(h = 0, lty = 2)
```


P−t − P−c
date