Udacity – AB Testing – Final Project

Python Code
https://www.kaggle.com/tammyrotem/ab-tests-with-python
https://github.com/shubhamlal11/Udacity-AB-Testing-Final-Project/blob/master/AB_testing_project.ipynb
https://github.com/baumanab/udacity_ABTesting#summary

R Code
https://rstudio-pubs-static.s3.amazonaws.com/347758_9da9522d18a8455fb810c48b11ff9824.html

Project Description
https://docs.google.com/document/u/1/d/1aCquhIqsUApgsxQ8-SQBAigFDcfWVVohLEXcV6jWbdI/pub?embedded=True

Udacity tested a change where if the student clicked "start free trial", they were asked how much time they had available to devote to the course. If the student indicated 5 or more hours per week, they would be taken through the checkout process as usual. If they indicated fewer than 5 hours per week, a message would appear indicating that Udacity courses usually require a greater time commitment for successful completion, and suggesting that the student might like to access the course materials for free. At this point, the student would have the option to continue enrolling in the free trial, or access the course materials for free instead.

The hypothesis was that this might set clearer expectations for students upfront, thus reducing the number of frustrated students who left the free trial because they didn't have enough time—without significantly reducing the number of students to continue past the free trial and eventually complete the course.

The unit of diversion is a cookie, although if the student enrolls in the free trial, they are tracked by user-id from that point forward. The same user-id cannot enroll in the free trial twice. For users that do not enroll, their user-id is not tracked in the experiment, even if they were signed in when they visited the course overview page.

Metric Choice
- Use it as a “invariant metric” or “evaluation metric”?
- Any place "unique cookies" are mentioned, the uniqueness is determined by day. (That is, the same cookie visiting on different days would be counted twice.) User-ids are automatically unique since the site does not allow the same user-id to enroll twice.

Possible Options:
- Number of cookies: That is, number of unique cookies to view the course overview page. (dmin=3000)
- Number of user-ids: That is, number of users who enroll in the free trial. (dmin=50)
- Number of clicks: That is, number of unique cookies to click the "Start free trial" button (which happens before the free trial screener is trigger). (dmin=240)
- Click-through-probability: That is, number of unique cookies to click the "Start free trial" button divided by number of unique cookies to view the course overview page. (dmin=0.01)
- Gross conversion: That is, number of user-ids to complete checkout and enroll in the free trial divided by number of unique cookies to click the "Start free trial" button. (dmin= 0.01)
- Retention: That is, number of user-ids to remain enrolled past the 14-day boundary (and thus make at least one payment) divided by number of user-ids to complete checkout. (dmin=0.01)
- Net conversion: That is, number of user-ids to remain enrolled past the 14-day boundary (and thus make at least one payment) divided by the number of unique cookies to click the "Start free trial" button. (dmin= 0.0075)

3.1 Invariate Metrics - Sanity Checks

Metric Name	Metric Formula	$Dmin$	Notation
Number of Cookies in Course Overview Page	# unique daily cookies on page	3000 cookies	C_k
Number of Clicks on Free Trial Button	# unique daily cookies who clicked	240 clicks	C_l
Free Trial button Click-Through-Probability	$\frac{C_l}{C_k}$	0.01	CTP

3.2 Evaluation Metrics - Performance Indicators

Metric Name	Metric Formula	$Dmin$	Notation
Gross Conversion	$\frac{enrolled}{C_l}$	0.01	$Conversion_{Gross}$
Retention	$\frac{paid}{enrolled}$	0.01	$Retention$
Net Conversion	$\frac{paid}{C_l}$	0.0075	$Conversion_{Net}$

Measuring Variability

Unique cookies to view course overview page per day:	40000
Unique cookies to click "Start free trial" per day:	3200
Enrollments per day:	660
Click-through-probability on "Start free trial":	0.08
Probability of enrolling, given click:	0.20625
Probability of payment, given enroll:	0.53
Probability of payment, given click	0.1093125

```
import math as mt
import numpy as np
import pandas as pd
from scipy.stats import norm
```

```
#Let's place this estimators into a dictionary for ease of use later
baseline = {"Cookies":40000,"Clicks":3200,"Enrollments":660,"CTP":0.08,"GConversion":0.20625,
            "Retention":0.53,"NConversion":0.109313}
```

Assuming a sample size of 5,000 cookies visiting the course overview page per day (as given in project's instructions) - we want to estimate a standard deviation, for the evaluation metrics only. The sample size we are considering should be smaller than the "population" we collected and small enough to have two groups with that size.

For all the calculations to follow we need to scale our collected counts estimates of metrics with the sample size we specified for variance estimation. In this case, from 40000 unique cookies to visit the course overview page per day, to 5000.

```
#Scale The counts estimates
baseline["Cookies"] = 5000
baseline["Clicks"]=baseline["Clicks"]*(5000/40000)
baseline["Enrollments"]=baseline["Enrollments"]*(5000/40000)
baseline
```

```
>>> {'CTP': 0.08,
 'Clicks': 400.0,
 'Cookies': 5000,
 'Enrollments': 82.5,
 'GConversion': 0.20625,
 'NConversion': 0.109313,
 'Retention': 0.53}
```

In order to estimate variance analytically, we can assume metrics which are probabilities ($\hat{p}$) are binomially distributed, so we can use this formula for the standard deviation:

$$SD = \sqrt{\frac{\hat{p}*(1-\hat{p})}{n}}$$

This assumption is only valid when the **unit of diversion** of the experiment is equal to the **unit of analysis** (the denominator of the metric formula). In the cases when this is not valid, the actual variance might be different and it is recommended to estimate it empirically.

For each metric, we need to plug two variables into the formula:
$\hat{p}$ - baseline probability of the event to occur
n - sample size

- **Gross Conversion** - The baseline probability for Gross Conversion can be calculated by the number of users to enroll in a free trial divided by the number of cookies clicking the free trial. In other words, the probability of enrollment given a click. In this case, the unit of diversion (Cookies), that is the element by which we differentiate samples and assign them to control and experiment groups, is equall to the unit of analysis (cookies who click), that is the denominator of the formula to calculate Gross Conversion (GC). When this is the case, this analytic estimate of variance is sufficient.

```
# Let's get the p and n we need for Gross Conversion (GC)
# and compute the Standard Deviation(sd) rounded to 4 decimal digits.
GC={}
GC["d_min"]=0.01
GC["p"]=baseline["GConversion"]
#p is given in this case - or we could calculate it from enrollments/clicks
GC["n"]=baseline["Clicks"]
GC["sd"]=round(mt.sqrt((GC["p"]*(1-GC["p"]))/GC["n"]),4)
GC["sd"]
```

>> 0.0202

- **Retention** - The baseline probability for retention is the number of paying users (enrolled after 14 free days) divided by the number of total enrolled users. In other words, the probability of payment, given enrollment. The sample size is the number of enrolled users. In this case, unit of diversion is not equal to unit of analysis (users who enrolled) so an analytical estimation is not enough - if we had the data for these estimates, we would want to estimate this variance empirically as well.

In [5]:
```
# Let's get the p and n we need for Retention(R)
# and compute the Stansard Deviation(sd) rounded to 4 decimal digits.
R={}
R["d_min"]=0.01
R["p"]=baseline["Retention"]
R["n"]=baseline["Enrollments"]
R["sd"]=round(mt.sqrt((R["p"]*(1-R["p"]))/R["n"]),4)
R["sd"]
```

Out[5]:
```
0.0549
```

- **Net Conversion** - The baseline probability for the net conversion is the number of paying users divided by the number of cookies that clicked the free trial button. In other words, the probability of payment, given a click. The sample size is the number of cookies that clicked. In this case, the unit of analysis and diversion are equal so we expect a good enough estimation analytically.

```
# Let's get the p and n we need for Net Conversion (NC)
# and compute the Standard Deviation (sd) rounded to 4 decimal digits.
NC={}
NC["d_min"]=0.0075
NC["p"]=baseline["NConversion"]
NC["n"]=baseline["Clicks"]
NC["sd"]=round(mt.sqrt((NC["p"]*(1-NC["p"]))/NC["n"]),4)
NC["sd"]
```

```
0.0156
```

Sizing & Duration

Choosing Number of Samples given Power. Using the analytic estimates of variance, how many page views total (across both groups) would you need to collect to adequately power the experiment? Use an alpha (significance level) of 0.05 and a beta (1 - power) of 0.2. Make sure you have enough power for each metric.

The minimum sample size for control and experiment groups, which provides probability of Type I Error α, Power $1-\beta$, detectable effect d and baseline conversion rate p (simple hypothesis $H_0 : P_{cont} - P_{exp} = 0$ against simple alternative $H_A : P_{cont} - P_{exp} = d$ is:

$$n = \frac{\left(Z_{1-\frac{\alpha}{2}}\, sd_1 + Z_{1-\beta} sd_2\right)^2}{d^2}, \text{ with:}$$

$$sd_1 = \sqrt{p(1-p) + p(1-p)}$$

$$sd_2 = \sqrt{p(1-p) + (p+d)(1-(1-(p+d))}$$

Now, let's break down what inputs we need and which calculations still need to be made. Regarding inputs, we have all the data we need: Type 1 error (α), power ($1-\beta$), detectable change ($d = D_{min}$) and baseline conversion rate (our $\hat{p}$). What we need to calculate:

- Get Z score for $1-\frac{\alpha}{2}$ and for $1-\beta$
- Get standard deviations 1 & 2, that is for both the baseline and for expected changed rate All these components will finally yield the number we need.

* Get z-score critical value and Standard Deviations

Use python's scipy.stats.norm package to get all the required methods for normal distribution. The ppf method gives us access to the Percent Point Function (ppf) or Quantile Function, and besides it being the inverse of the Cummulative Distribution Function (cdf), this is the functions that will give back our required critical z-score.

```
#Inputs: required alpha value (alpha should already fit the required test)
#Returns: z-score for given alpha
def get_z_score(alpha):
    return norm.ppf(alpha)

# Inputs p-baseline conversion rate which is our estimated p and d-minimum detectable change
# Returns
```

```
def get_sds(p,d):
    sd1=mt.sqrt(2*p*(1-p))
    sd2=mt.sqrt(p*(1-p)+(p+d)*(1-(p+d)))
    sds=[sd1,sd2]
    return sds

# Inputs:sd1-sd for the baseline,sd2-sd for the expected change,alpha,beta,d-d_min,p-baseline estimate p
# Returns: the minimum sample size required per group according to metric denominator
def get_sampSize(sds,alpha,beta,d):
    n=pow((get_z_score(1-alpha/2)*sds[0]+get_z_score(1-beta)*sds[1]),2)/pow(d,2)
    return n
```

```
GC["d"]=0.01
R["d"]=0.01
NC["d"]=0.0075
```

- **Gross Conversion**

```
# Let's get an integer value for simplicity
GC["SampSize"]=round(get_sampSize(get_sds(GC["p"],GC["d"]),0.05,0.2,GC["d"]))
GC["SampSize"]
```

```
25835.0
```

This means we need at least 25,835 cookies who click the Free Trial button - per group! That means that if we got 400 clicks out of 5000 pageviews (`400/5000 = 0.08`) -> So, we are going to need `GC["SampSize"]/0.08 = 322,938` pageviews, again ; per group! Finally, the total amount of samples per the Gross Conversion metric is:

```
GC["SampSize"]=round(GC["SampSize"]/0.08*2)
GC["SampSize"]
```

```
645875.0
```

- Retention

```
# Getting a nice integer value
R["SampSize"]=round(get_sampSize(get_sds(R["p"],R["d"]),0.05,0.2,R["d"]))
R["SampSize"]
```

```
39087.0
```

This means that we need 39,087 users who enrolled per group! We have to first convert this to cookies who clicked, and then to cookies who viewed the page, then finally to multipky by two for both groups.

```
R["SampSize"]=R["SampSize"]/0.08/0.20625*2
R["SampSize"]
```

```
4737818.1818181816
```

This takes us as high as over 4 million page views total, this is practically impossible because we know we get about 40,000 a day, this would take well over 100 days. This means we have to drop this metric and not continue to work with it because results from our experiment (which is much smaller) will be biased.

- Net Conversion

```
# Getting a nice integer value
NC["SampSize"]=round(get_sampSize(get_sds(NC["p"],NC["d"]),0.05,0.2,NC["d"]))
NC["SampSize"]
```

```
27413.0
```

So, needing 27,413 cookies who click per group takes us all the way up to:

```
NC["SampSize"]=NC["SampSize"]/0.08*2
NC["SampSize"]
```

```
685325.0
```

We are all the way up to 685,325 cookies who view the page. This is more than what was needed for Gross Conversion, so this will be our number. Assuming we take 80% of each days pageviews, the data collection period for this experiment (the period in which the experiment is revealed) will be about 3 weeks.

Analysis

Data Column Descriptions:

- Pageviews: Number of unique cookies to view the course overview page that day.
- Clicks: Number of unique cookies to click the course overview page that day.
- Enrollments: Number of user-ids to enroll in the free trial that day.
- Payments: Number of user-ids who who enrolled on that day to remain enrolled for 14 days and thus make a payment. (Note that the date for this column is the start date, that is, the date of enrollment, rather than the date of the payment. The payment happened 14 days later. Because of this, the enrollments and payments are tracked for 14 fewer days than the other columns.)

```
# we use pandas to load datasets
control=pd.read_csv("../input/control-data/control_data.csv")
experiment=pd.read_csv("../input/experiment-data/experiment_data.csv")
control.head()
```

	Date	Pageviews	Clicks	Enrollments	Payments
0	Sat, Oct 11	7723	687	134.0	70.0
1	Sun, Oct 12	9102	779	147.0	70.0
2	Mon, Oct 13	10511	909	167.0	95.0
3	Tue, Oct 14	9871	836	156.0	105.0
4	Wed, Oct 15	10014	837	163.0	64.0

* Sanity Checks

Start by checking whether your invariant metrics are equivalent between the two groups. If the invariant metric is a simple count that should be randomly split between the 2 groups, you can use a binomial test. Otherwise, you will need to construct a confidence interval for a difference in proportions, then check whether the difference between group values falls within that confidence level.

We have 3 Invariant metrics:

- Number of Cookies in Course Overview Page
- Number of Clicks on Free Trial Button
- Free Trial button Click-Through-Probability

Two of these metrics are simple counts like number of cookies or number of clicks and the third is a probability (CTP). We will use two different ways of checking whether these obsereved values are like we expect (if in fact the experiment was not damaged.

- Number of cookies who viewed the course overview page

- Starting from this simple invariant metric, we want to count the total amount of cookie page views we diverted to each group and see if there is a significant difference in the amount of cookies. A significant difference will imply a biased experiment that we should not rely on it's results.

```
pageviews_cont=control['Pageviews'].sum()
pageviews_exp=experiment['Pageviews'].sum()
pageviews_total=pageviews_cont+pageviews_exp
print ("number of pageviews in control:", pageviews_cont)
print ("number of Pageviewsin experiment:" ,pageviews_exp)
```

>>> number of pageviews in control: 345543
number of Pageviewsin experiment: 344660

Ok so these look like pretty close numbers. Now, let's make sure this difference in amounts is not significant and is random and even like we expected. We can model this diversion in the following way:
We expect the amount of pageviews in the control group to be about a half (50%) of the total pageviews in both groups, so we can define a random variable with an easy to use distribution.
A binomial random variable will be the number of successes we can expect to get out of N experiments, given the probability of a single success. So, if we consider being assigned to a group (control, for example) a success with probability 0.5 (random!), the number of samples which get assigned to the group is the value of our random binomial variable!

This get's easier thanks to the central limit theorem which let's us approximate the binomial distribution to a normal distribution (when n is large enough) with a mean of p and a standard deviation $\sqrt{\frac{p(1-p)}{N}}$

$$X \sim N(p, \sqrt{\frac{p(1-p)}{N}})$$

What we want to test is whether our observed $\hat{p}$ (number of samples in control divided by total number of damples in both groups) is not significantly different than $p = 0.5$. In order to do that we can calculate the margin of error acceptable at a 95% confidence level:

$$ME = Z_{1-\frac{\alpha}{2}} SD$$

Finally, a confidence interval can be derived to tell us in which range an observed p can exist and be acceptable as "the same" as the expected value.

$$CI = [\hat{p} - ME, \hat{p} + ME]$$

When our obsereved $\hat{p}$ is within this range, all is well and the test was passed.

```
p=0.5
alpha=0.05
p_hat=round(pageviews_cont/(pageviews_total),4)
sd=mt.sqrt(p*(1-p)/(pageviews_total))
ME=round(get_z_score(1-(alpha/2))*sd,4)
print ("The confidence interval is between",p-ME,"and",p+ME,"; Is",p_hat,"inside this range?")
```

>>> The confidence interval is between 0.4988 and 0.5012 ; Is 0.5006 inside this range?
Our observed p^ is inside this range which means the difference in number of samples between groups is expected. So far so good, since this invariant metric sanity test passes!

- Number of cookies who clicked the Free Trial Button We are going to address this count with the same strategy as before.

```
clicks_cont=control['Clicks'].sum()
clicks_exp=experiment['Clicks'].sum()
clicks_total=clicks_cont+clicks_exp

p_hat=round(clicks_cont/clicks_total,4)
sd=mt.sqrt(p*(1-p)/clicks_total)
ME=round(get_z_score(1-(alpha/2))*sd,4)
print ("The confidence interval is between",p-ME,"and",p+ME,"; Is",p_hat,"inside this range?")
```

>>> The confidence interval is between 0.4959 and 0.5041 ; Is 0.5005 inside this range?
We have another pass! Great, so far it still seems all is well with our experiment results. Now, for the final metric which is a probability.

- **Click-through-probability of the Free Trial Button** In this case, we want to make sure the proportion of clicks given a pageview (our observed CTP) is about the same in both groups (since this was not expected to change due to the experiment). In order to check this out we will calculate the CTP in each group and calculate a confidence interval for the expected difference between them.

In other words, we expect to see no difference ($CTP_{exp} - CTP_{cont} = 0$), with an acceptable margin of error, dictated by our calculated confidence interval. The changes we should notice are for the calculation of the standard error - which in this case is a pooled standard error.

$$SD_{pool} = \sqrt{\hat{p_{pool}}(1 - \hat{p_{pool}}(\frac{1}{N_{cont}} + \frac{1}{N_{exp}})}$$

with

$$\hat{p_{pool}} = \frac{x_{cont} + x_{exp}}{N_{cont} + N_{exp}}$$

We should understand that CTP is a proportion in a population (amount of events x in a population n) like the amount of clicks out of the amount of pageviews..

```
ctp_cont=clicks_cont/pageviews_cont
ctp_exp=clicks_exp/pageviews_exp
d_hat=round(ctp_exp-ctp_cont,4)
p_pooled=clicks_total/pageviews_total
sd_pooled=mt.sqrt(p_pooled*(1-p_pooled)*(1/pageviews_cont+1/pageviews_exp))
ME=round(get_z_score(1-(alpha/2))*sd_pooled,4)
print ("The confidence interval is between",0-ME,"and",0+ME,"; Is",d_hat,"within this range?")
```

```
The confidence interval is between -0.0013 and 0.0013 ; Is 0.0001 within this range?
```

Check for Practical and Statistical Significance

Next, for your evaluation metrics, calculate a confidence interval for the difference between the experiment and control groups, and check whether each metric is statistically and/or practically significance. A metric is statistically significant if the confidence interval does not include 0 (that is, you can be confident there was a change), and it is practically significant if the confidence interval does not include the practical significance boundary (that is, you can be confident there is a change that matters to the business.)

If you have chosen multiple evaluation metrics, you will need to decide whether to use the Bonferroni correction. When deciding, keep in mind the results you are looking for in order to launch the experiment. Will the fact that you have multiple metrics make those results more likely to occur by chance than the alpha level of 0.05?

- **Gross Conversion** A metric is statistically significant if the confidence interval does not include 0 (that is, you can be confident there was a change), and it is practically significant if the confidence interval does not include the practical significance boundary (that is, you can be confident there is a change that matters to the business.)

> **Important:** The given spreadsheet lists pageviews and clicks for 39 days, while it only lists enrollments and payments for 23 days. So, when working with enrollments and payments we should notice using only the corresponding pageviews and clicks, and not all of them.

```
# Count the total clicks from complete records only
clicks_cont=control["Clicks"].loc[control["Enrollments"].notnull()].sum()
clicks_exp=experiment["Clicks"].loc[experiment["Enrollments"].notnull()].sum()
```

```
#Gross Conversion - number of enrollments divided by number of clicks
enrollments_cont=control["Enrollments"].sum()
enrollments_exp=experiment["Enrollments"].sum()

GC_cont=enrollments_cont/clicks_cont
GC_exp=enrollments_exp/clicks_exp
GC_pooled=(enrollments_cont+enrollments_exp)/(clicks_cont+clicks_exp)
GC_sd_pooled=mt.sqrt(GC_pooled*(1-GC_pooled)*(1/clicks_cont+1/clicks_exp))
GC_ME=round(get_z_score(1-alpha/2)*GC_sd_pooled,4)
GC_diff=round(GC_exp-GC_cont,4)
print("The change due to the experiment is",GC_diff*100,"%")
print("Confidence Interval: [",GC_diff-GC_ME,",",GC_diff+GC_ME,"]")
print ("The change is statistically significant if the CI doesn't include 0. In that case, it i
s practically significant if",-GC["d_min"],"is not in the CI as well.")
```

>>> The change due to the experiment is -2.06 %
Confidence Interval: [-0.0292 , -0.012]
The change is statistically significant if the CI doesn't include 0. In that case, it is practically significant if -0.01 is not in the CI as well.

According to this result there was a change due to the experiment, that change was both statistically and practically significant. We have a negative change of 2.06%, when we were willing to accept any change greater than 1%. This means the Gross Conversion rate of the experiment group (the one exposed to the change, i.e. asked how many hours they can devote to studying) has decreased as expected by 2% and this change was significant. This means less people enrolled in the Free Trial after due to the pop-up.

- **Net Conversion** The hypothesis is the same as before just with net conversion instead of gross. At this point we expect the fraction of payers (out of the clicks) to decrease as well.

```
#Net Conversion - number of payments divided by number of clicks
payments_cont=control["Payments"].sum()
payments_exp=experiment["Payments"].sum()

NC_cont=payments_cont/clicks_cont
NC_exp=payments_exp/clicks_exp
NC_pooled=(payments_cont+payments_exp)/(clicks_cont+clicks_exp)
NC_sd_pooled=mt.sqrt(NC_pooled*(1-NC_pooled)*(1/clicks_cont+1/clicks_exp))
NC_ME=round(get_z_score(1-alpha/2)*NC_sd_pooled,4)
NC_diff=round(NC_exp-NC_cont,4)
print("The change due to the experiment is",NC_diff*100,"%")
print("Confidence Interval: [",NC_diff-NC_ME,",",NC_diff+NC_ME,"]")
print ("The change is statistically significant if the CI doesn't include 0. In that case, it i
s practically significant if",NC["d_min"],"is not in the CI as well.")
```

```
The change due to the experiment is -0.49 %
Confidence Interval: [ -0.0116 , 0.0018 ]
The change is statistically significant if the CI doesn't include 0. In that case, it is practi
cally significant if 0.0075 is not in the CI as well.
```

In this case we got a change size of less than a 0.5%, a very small decrease which is not statistically significant, and as such not practically significant.

Run Sign Tests (Double Check)

For each evaluation metric, do a sign test using the day-by-day breakdown. If the sign test does not agree with the confidence interval for the difference, see if you can figure out why.

In a sign test we get another angle at analyzing the results we got - we check if the trend of change we observed (increase or decrease) was evident in the daily data. We are goint to compute the metric's value per day and then count on how many days the metric was lower in the experiment group and this will be the number of succssesses for our binomial variable. Once this is defined we can look at the proportion of days of success out of all the available days.

```
#let's first create the dataset we need for this:
# start by merging the two datasets
full=control.join(other=experiment,how="inner",lsuffix="_cont",rsuffix="_exp")
#Let's look at what we got
full.count()
```

```
Date_cont           37
Pageviews_cont      37
Clicks_cont         37
Enrollments_cont    23
Payments_cont       23
Date_exp            37
Pageviews_exp       37
Clicks_exp          37
Enrollments_exp     23
Payments_exp        23
dtype: int64
```

```
#now we only need the complete data records
full=full.loc[full["Enrollments_cont"].notnull()]
full.count()
```

```
Date_cont           23
Pageviews_cont      23
Clicks_cont         23
Enrollments_cont    23
Payments_cont       23
Date_exp            23
Pageviews_exp       23
Clicks_exp          23
Enrollments_exp     23
Payments_exp        23
dtype: int64
```

```
# Perfect! Now, derive a new column for each metric, so we have it's daily values
# We need a 1 if the experiment value is greater than the control value=
x=full['Enrollments_cont']/full['Clicks_cont']
y=full['Enrollments_exp']/full['Clicks_exp']
full['GC'] = np.where(x<y,1,0)
# The same now for net conversion
z=full['Payments_cont']/full['Clicks_cont']
w=full['Payments_exp']/full['Clicks_exp']
full['NC'] = np.where(z<w,1,0)
full.head()
```

	Date_cont	Pageviews_cont	Clicks_cont	Enrollments_cont	Payments_cont	Date_exp	Pageviews_exp	Clicks_exp	Enrollments_exp
0	Sat, Oct 11	7723	687	134.0	70.0	Sat, Oct 11	7716	686	105.0
1	Sun, Oct 12	9102	779	147.0	70.0	Sun, Oct 12	9288	785	116.0
2	Mon, Oct 13	10511	909	167.0	95.0	Mon, Oct 13	10480	884	145.0
3	Tue, Oct 14	9871	836	156.0	105.0	Tue, Oct 14	9867	827	138.0
4	Wed, Oct 15	10014	837	163.0	64.0	Wed, Oct 15	9793	832	140.0

```
GC_x=full.GC[full["GC"]==1].count()
NC_x=full.NC[full["NC"]==1].count()
n=full.NC.count()
print("No. of cases for GC:",GC_x,'\n',
      "No. of cases for NC:",NC_x,'\n',
      "No. of total cases",n)
```

```
No. of cases for GC: 4
 No. of cases for NC: 10
 No. of total cases 23
```

We can forget all about this part and just use an online sign test calculator, but for me that is just no fun - so I will implement the calculations behind it.
What we want to do after we count the amount of days in which the experiment group had a higher metric value than that of the control group, is to see if that number is likely to be seen again in a new experiment (significance). We assume the chance of a day like this is random (50% chance to happen) and then use the binomial distribution with $p = 0.5$ and the number of experiments (days) to tell us the probability of this happening according to a random chance.
So, according to the binomial distribution with $p = 0.5$ and n =total number of days; we want to now the probability of x days being a success (higher metric value in experiment). Because we are doing a two-tailed test we want to double this probability and once we have we can call it the $p - value$ and compare it to our α. If the $p - value$ is greater than the α the result is not significant and vice-versa.

$$p(successes) = \frac{n!}{x!(n-x)!}p^x(1-p)^{n-x}$$

Recall that a $p - value$ is the probability of observing a test statistic as or more extreme than that observed. If we observed 2 days like that, the $p - value$ for the test is: $p - value = P(x <= 2)$. We only need to remember the following:

$$P(x <= 2) = P(0) + P(1) + P(2).$$

https://www.graphpad.com/quickcalcs/binomial1/
http://sphweb.bumc.bu.edu/otlt/mph-modules/bs/bs704_nonparametric/BS704_Nonparametric5.html

```
#first a function for calculating probability of x=number of successes
def get_prob(x,n):
    p=round(mt.factorial(n)/(mt.factorial(x)*mt.factorial(n-x))*0.5**x*0.5**(n-x),4)
    return p
#next a function to compute the pvalue from probabilities of maximum x
def get_2side_pvalue(x,n):
    p=0
    for i in range(0,x+1):
        p=p+get_prob(i,n)
    return 2*p
```

Finally, to conduct the sign test itself: we will calculate the p-value for each metric, using the counts GC_x,NC_x and n and the function *get_2side_pvalue*.

```
print ("GC Change is significant if",get_2side_pvalue(GC_x,n),"is smaller than 0.05")
print ("NC Change is significant if",get_2side_pvalue(NC_x,n),"is smaller than 0.05")
```

>>>C Change is significant if 0.0026 is smaller than 0.05
NC Change is significant if 0.6774 is smaller than 0.05

We get the same conclusions as we got from our effect size calculation: the change in Gross conversion was indeed significant, while the change in Net conversion was not.

Make a Recommendation

Finally, make a recommendation. Would you launch this experiment, not launch it, dig deeper, run a follow-up experiment, or is it a judgment call? If you would dig deeper, explain what area you would investigate. If you would run follow-up experiments, briefly describe that experiment. If it is a judgment call, explain what factors would be relevant to the decision.

➔ At this point, once we have seen that the actual underlying goal we had was not reached (increase fraction of paying users by asking them in advance if they have the time to invest in the course), we can only recommend to not continue with change. It may have caused a change in Gross conversion, but it didn't for net conversion.

➔This experiment was designed to determine whether filtering students as a function of study time commitment would improve the overall student experience and the coaches' capacity to support students who are likely to complete the course, without significantly reducing the number of students who continue past the free trial. A statistically and practically signficant decrease in Gross Conversion was observed but with no significant differences in Net Conversion. This translates to a decrease in enrollment not coupled to an increase

in students staying for the requisite 14 days to trigger payment. Considering this, my recomendation is not to launch, but rather to pursue other experiments.

Follow-Up Experiment: How to Reduce Early Cancellations
If you wanted to reduce the number of frustrated students who cancel early in the course, what experiment would you try? Give a brief description of the change you would make, what your hypothesis would be about the effect of the change, what metrics you would want to measure, and what unit of diversion you would use. Include an explanation of each of your choices.
➔The construct of student frustration could be assigned an operational definition of "cancel early," where a convenient definition and measure of early cancellation is prior to the end of the 14 day trial period in which payment is triggered. An early cancellation is not necessarily indicative of frustration but could be from other causes, such as a course not being aligned to the students needs or expectations in terms of content. For preventng early cancellation there are two primary logical timepoint opportunities for intervention, (1) pre-enrollment, and (2) post-enrollment but pre-payment.

The first opportunity for intervention was explored above wherein a poll regarding time commitment was used as to filter out students likely to become frustrated. This filter focused only on time commitment to the class and did not address other reasons why a student might become frustrated and cancel early. Even if the student was sincere in their response and dilligent in their study, they may become frustrated if they don't have the suggested pre-requisite skills and experience. That is, their committed time may not be enough if they don't come in with the pre-requisite skill set. Adding a checklist of pre-requisite skills to the popup regarding time commitment may be informative. This experiment would leverage the infrastrucure and data pipeline of the original experiment and be set up in the same way as the original, including the unit of diversion. The only difference would be the information in the form. If the student's answer meets the time and pre-requisite requirements (radiobox checklist) they are directed to enroll in the free trial, otherwise they are encouraged to access the free version. This experiment would be low cost in terms of resources and may increase the selectivity of the pre-enrollment filter. A succesful experiment would be one in which there is a signficant decrease in Gross Conversion coupled to a significant increase in Net Conversion.

A variety of approaches could be used to intervene post-enrollment but pre-payment and could be deployed concurrently with pre-enrollment intervention. An ideal approach would be one which minimizes the use of additional coaching resources to best meet the original intent of the intervention. An effective approach may be to employ peer coaching/guidance by means of team formation. If a student has a team of other students which they could consult, discuss coursework and frustrations with, and be accountable to, they may be more likely to stick out the growing pains and stay for the long term. The experiment would function in the following manner.

Setup: Upon enrollment students will either be randomly assigned to a control group in which they are not funnelled into a team, or an experiment group in which they are.

Null Hypothesis: Participation in a team will not increase the number of students enrolled beyond the 14 day free trial period by a significant amount.

Unit of Diversion: The unit of diversion will be user-id as the change takes place after a student creates an account and enrolls in a course.

Invariant Metrics: The invariant metric will be user-id, since an equal distribution between experiment and control would be expected as a property of the setup.

Evaluation Metrics: The evaluation metric willl be Retention. A statistically and practically significant increase in Retention would indicate that the change is succesful.

If a statistically and practically signifcant positive change in Retention is observed, assuming an acceptable impact

on overall Udacity resources (setting up and maintaining teams will require resource use), the experiment will be launched.