Free_Trial_Screener

December 31, 2021

1 Free Trail Screener Project

At the time of this experiment, Udacity courses currently have two options on the course overview page: **start free trial**, and **access course materials**.

If the student clicks "start free trial", they will be asked to enter their credit card information, and then they will be enrolled in a free trial for the paid version of the course. After 14 days, they will automatically be charged unless they cancel first.

If the student clicks "access course materials", they will be able to view the videos and take the quizzes for free, but they will not receive coaching support or a verified certificate, and they will not submit their final project for feedback.

In the experiment, 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. If this hypothesis held true, Udacity could improve the overall student experience and improve coaches' capacity to support students who are likely to 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.

1.1 Experiment Design

1.1.1 Metric Choice

List which metrics you will use as invariant metrics and evaluation metrics here.

For each metric, explain both why you did or did not use it as an invariant metric and why you did or did not use it as an evaluation metric. Also, state what results you will look for in your evaluation metrics in order to launch the experiment.

Invariant Metrics 1. Number of cookies

Number of unique cookies to see the course overview page. The unit of diversion and distribution amongst the control and experiment groups are expected to be the same because the setup of controlled and experimented groups assigns two similarily same number of cookies.

2. Number of clicks

Number of unique cookies to click the "Start free trial" button. Since users got asked how much time to complete a course before seeing the page, the unit of diversion between control and experiment groups are expected to be the same.

3. Click-through-probability

Number of unique cookies to click the "Start free trial" button divided by number of unique cookies to view the course overview page. It is expected to be the same because users see the change before clicking.

Evaluation Metrics 1. Gross conversion

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.

2. Retention

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.

3. Net conversion

Number of users who enroll in the free trial. After 14 days, users will determine to use it or not, so we could utilize it as our evaluation metric.

1.1.2 Measuring Standard Deviation

List the standard deviation of each of your evaluation metrics given a sample size of 5000 cookies.

For each of your evaluation metrics, indicate whether you think the analytic estimate would be comparable to the the empirical variability, or whether you expect them to be different (in which case it might be worth doing an empirical estimate if there is time). Briefly give your reasoning in each case.

Metrics	Baseline value
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

[4]:
```
def SE(prob, size):
    return ((prob * (1-prob))/size)**0.5
```

[5]:
```
# gross conversion
prob_enroll = 0.20625
size_gross_conversion = 5000 * 0.08
# retention
prob_payment_enroll = 0.53
size_retention = 5000 * 0.08 * 0.20625
# net conversion
prob_conversion = 0.1093125
size_net_conversion = 5000 * 0.08

print("SE of gross conversion", SE(prob_enroll, size_gross_conversion))
print("SE of retention", SE(prob_payment_enroll, size_retention))
print("SE of net conversion", SE(prob_conversion, size_net_conversion))
```

```
SE of gross conversion 0.020230604137049392
SE of retention 0.05494901217850908
SE of net conversion 0.01560154458248846
```

1.1.3 Sizing

Indicate whether you will use the Bonferroni correction during your analysis phase, and give the number of pageviews you will need to power you experiment appropriately.

Bonferroni Correction

[20]:
```
import scipy.stats as st
alpha_overall = 0.05
alpha_bonferroni = alpha_overall / 3
st.norm.ppf(1-alpha_bonferroni/2)
```

[20]:
```
2.3939797998185104
```

- 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)

[21]:
```
print("Margin Error of gross conversion", 2.3939797998185104 * SE(prob_enroll,␣
 ↪size_gross_conversion))
print("Margin Error of retention", 2.3939797998185104 * SE(prob_payment_enroll,␣
 ↪size_retention))
print("Margin Error of net conversion", 2.3939797998185104 * SE(prob_conversion,␣
 ↪size_net_conversion))
```

```
Margin Error of gross conversion 0.04843165764222103
Margin Error of retention 0.13154682517533206
Margin Error of net conversion 0.03734978257644529
```

By caculating confidence intervals of dmin, we could see some metric contain 0. So it is not significant. We won't use Bonferroni Correction right now.

Choosing Number of Samples given Power

Using the analytic estimates of variance, how many pageviews total (across both groups) would you need to collect to adequately power the experiment? Use an alpha of 0.05 and a beta of 0.2. Make sure you have enough power for each metric.

We could use online sample size calculator. (https://www.evanmiller.org/ab-testing/sample-size.html)

Gross Conversion:
Baseline Conversion Rate: 0.20625 / Minimum Detectable Effect: 0.01 -> Sample Size: 25835
Total Sample Size (Control&Experiment): 25835 * 2 = 51670 (clicks)
Pageviews: 51670/0.08 = 645875

Retention:
Baseline Conversion Rate: 0.53 / Minimum Detectable Effect: 0.01 -> Sample Size: 39115
Total Sample Size (Control&Experiment): 39115 * 2 = 78230 (enrolls)
Pageviews: 78230/(660/40000) = 4741212

Net Conversion: Baseline Conversion Rate: 0.1093125 / Minimum Detectable Effect: 0.0075 -> Sample Size: 27413
Total Sample Size (Control&Experiment): 27413 * 2 = 54826 (clicks)
Pageviews: 54826/0.08 = 685325

We will design functions to get the sample size. (Source:https://stats.stackexchange.com/questions/392979/ab-test-sample-size-calculation-by-hand)

[62]:
```python
# utilities to get sample size
from scipy.stats import norm
import numpy as np
def get_sample_size(alpha, beta, baseline, detectable_effect):
    get_alpha_ppf = norm.ppf(alpha/2)
    get_beta_ppf = norm.ppf(beta)
    p1 = baseline
    p2 = p1 + detectable_effect
    left_part = get_alpha_ppf * np.sqrt(2 * p1 * (1-p1))
    right_part = get_beta_ppf * np.sqrt(p1 * (1-p1)+ p2 * (1-p2))
    nominator = (left_part + right_part) ** 2
    denominator = (p2 - p1) ** 2
    return nominator/denominator
```

[64]:
```python
# gross conversion
get_sample_size(0.05, 0.2, 0.20625, 0.01)
```

[64]: 25834.700007480867

[65]:
```
# retention
get_sample_size(0.05, 0.2, 0.53, 0.01)
```

[65]: 39086.60966243208

[66]:
```
# Net Conversion
get_sample_size(0.05, 0.2, 0.1093125 , 0.0075)
```

[66]: 27413.33789636357

1.1.4 Duration vs. Exposure

What percentage of Udacity's traffic would you divert to this experiment (assuming there were no other experiments you wanted to run simultaneously)? Is the change risky enough that you wouldn't want to run on all traffic?

Given the percentage you chose, how long would the experiment take to run, using the analytic estimates of variance? If the answer is longer than a few weeks, then this is unreasonably long, and you should reconsider an earlier decision.

We use 100% percentage of traffic to run this experiment.

Pageviews	Days
Gross Conversion	16.14
Retention	118.5
Net Conversion	17.13

1.2 Experiment Analysis

[67]:
```
dates=['Sat, Oct 11', 'Sun, Oct 12', 'Mon, Oct 13', 'Tue, Oct 14',
       'Wed, Oct 15', 'Thu, Oct 16', 'Fri, Oct 17', 'Sat, Oct 18',
       'Sun, Oct 19', 'Mon, Oct 20', 'Tue, Oct 21', 'Wed, Oct 22',
       'Thu, Oct 23', 'Fri, Oct 24', 'Sat, Oct 25', 'Sun, Oct 26',
       'Mon, Oct 27', 'Tue, Oct 28', 'Wed, Oct 29', 'Thu, Oct 30',
       'Fri, Oct 31', 'Sat, Nov 1', 'Sun, Nov 2', 'Mon, Nov 3',
       'Tue, Nov 4', 'Wed, Nov 5', 'Thu, Nov 6', 'Fri, Nov 7',
       'Sat, Nov 8', 'Sun, Nov 9', 'Mon, Nov 10', 'Tue, Nov 11',
       'Wed, Nov 12', 'Thu, Nov 13', 'Fri, Nov 14', 'Sat, Nov 15',
       'Sun, Nov 16']
pageviews_cont=[ 7723,  9102, 10511,  9871, 10014,  9670,  9008,  7434,  8459,
       10667, 10660,  9947,  8324,  9434,  8687,  8896,  9535,  9363,
        9327,  9345,  8890,  8460,  8836,  9437,  9420,  9570,  9921,
        9424,  9010,  9656, 10419,  9880, 10134,  9717,  9192,  8630,
        8970]
pageviews_exp=[ 7716,  9288, 10480,  9867,  9793,  9500,  9088,  7664,  8434,
       10496, 10551,  9737,  8176,  9402,  8669,  8881,  9655,  9396,
```

```
        9262,  9308,  8715,  8448,  8836,  9359,  9427,  9633,  9842,
        9272,  8969,  9697, 10445,  9931, 10042,  9721,  9304,  8668,
        8988]
clicks_cont=[687, 779, 909, 836, 837, 823, 748, 632, 691, 861, 867, 838, 665,
       673, 691, 708, 759, 736, 739, 734, 706, 681, 693, 788, 781, 805,
       830, 781, 756, 825, 874, 830, 801, 814, 735, 743, 722]
clicks_exp=[686, 785, 884, 827, 832, 788, 780, 652, 697, 860, 864, 801, 642,
       697, 669, 693, 771, 736, 727, 728, 722, 695, 724, 789, 743, 808,
       831, 767, 760, 850, 851, 831, 802, 829, 770, 724, 710]
enrolls_cont=[134, 147, 167, 156, 163, 138, 146, 110, 131, 165, 196, 162, 127,
       220, 176, 161, 233, 154, 196, 167, 174, 156, 206]
enrolls_exp=[105, 116, 145, 138, 140, 129, 127,  94, 120, 153, 143, 128, 122,
       194, 127, 153, 213, 162, 201, 207, 182, 142, 182]
payment_cont=[ 70,  70,  95, 105,  64,  82,  76,  70,  60,  97, 105,  92,  56,
       122, 128, 104, 124,  91,  86,  75, 101,  93,  67]
payment_exp=[ 34,  91,  79,  92,  94,  61,  44,  62,  77,  98,  71,  70,  68,
        94,  81, 101, 119, 120,  96,  67, 123, 100, 103]
```

1.2.1 Sanity Checks

For each of your invariant metrics, give the 95% confidence interval for the value you expect to observe, the actual observed value, and whether the metric passes your sanity check.

For any sanity check that did not pass, explain your best guess as to what went wrong based on the day-by-day data. Do not proceed to the rest of the analysis unless all sanity checks pass.

[71]:
```
import math
def get_confidence_interval(num_control, num_experiment):
    sum_cont=sum(num_control)
    sum_exp=sum(num_experiment)
    SD=math.sqrt(0.5*0.5/(sum_cont+sum_exp))
    m=1.96*SD
    ci_min,ci_max=0.5-m,0.5+m
    print("Confidence Interval: [{},{}]".format(round(ci_min,4),round(ci_max,4)))
    print("Observed: ",round(sum_cont/(sum_exp+sum_cont),4))
    return
```

[72]:
```
# Pageviews
get_confidence_interval(pageviews_cont, pageviews_exp)
```

```
Confidence Interval: [0.4988,0.5012]
Observed:  0.5006
```

[73]:
```
# number of clicks
get_confidence_interval(clicks_cont, clicks_exp)
```

```
Confidence Interval: [0.4959,0.5041]
Observed:  0.5005
```

[81]:
```
# click_through_probability
ctp_cont = sum(clicks_cont)/sum(pageviews_cont)
ctp_exp = sum(clicks_exp)/sum(pageviews_exp)
d_hat = ctp_exp-ctp_cont
ctp_pool = (sum(clicks_cont)+sum(clicks_exp))/
 ↪(sum(pageviews_cont)+sum(pageviews_exp))
SE_ctp = math.sqrt(ctp_pool*(1-ctp_pool)*(1/sum(pageviews_cont)+1/
 ↪sum(pageviews_exp)))
m = 1.96 * SE_ctp
ci_min, ci_max = -m, m
print("Confidence Interval for ctp: [{},{}]".
 ↪format(round(ci_min,4),round(ci_max,4)))
print("Observed: ",round(d_hat,4))
```

```
Confidence Interval for ctp: [-0.0013,0.0013]
Observed:  0.0001
```

1.2.2 Result Analysis

Source:https://nancyyanyu.github.io/posts/8fdfc10f/

Effect Size Tests For each of your evaluation metrics, give a 95% confidence interval around the difference between the experiment and control groups. Indicate whether each metric is statistically and practically significant.

[84]:
```
# gross conversion
n = len(enrolls_exp)
d_min = 0.01
sum_clicks_cont = sum(clicks_cont[:n])
sum_clicks_exp = sum(clicks_exp[:n])
sum_enroll_cont = sum(enrolls_cont[:n])
sum_enroll_exp = sum(enrolls_exp[:n])

p_pool = (sum_enroll_exp + sum_enroll_cont)/(sum_clicks_exp + sum_clicks_cont)
SE_pool = math.sqrt(p_pool * (1 - p_pool) * (1/sum_clicks_exp + 1/
 ↪sum_clicks_cont))
m = SE_pool * 1.96
d_hat = sum_enroll_exp/sum_clicks_exp - sum_enroll_cont/sum_clicks_cont

print("Confidence Interval:[{},{}]".format(d_hat-m, d_hat+m))
print("Observed:", d_hat)
print ("Statistically significant:", d_hat + m < 0 or d_hat - m > 0 ,",  CI␣
 ↪doesn't include 0")
print("Practically significant:", True,",  CI doesn't include d_min or -d_min")
```

```
Confidence Interval:[-0.0291233583354044,-0.01198639082531873]
Observed: -0.020554874580361565
```

```
Statistically significant: True ,  CI doesn't include 0
Practically significant: True ,  CI doesn't include d_min or -d_min
```

[86]:
```
# retention
n = len(payment_exp)
d_min = 0.01
sum_payment_cont = sum(payment_cont[:n])
sum_payment_exp = sum(payment_exp[:n])
sum_enroll_cont = sum(enrolls_cont[:n])
sum_enroll_exp = sum(enrolls_exp[:n])
p_pool = (sum_payment_cont+sum_payment_exp) / (sum_enroll_cont+sum_enroll_exp)
SE_pool = math.sqrt(p_pool*(1-p_pool)*(1/sum_enroll_cont+1/sum_enroll_exp))
m=SE_pool*1.96
d_hat=sum_payment_exp/sum_enroll_exp-sum_payment_cont/sum_enroll_cont
print("Confidence Interval:[{},{}]".format(d_hat-m,d_hat+m))
print("Observed:",d_hat)
print ("Statistically significant:", d_hat+m<0 or d_hat-m>0 ,",  CI doesn't␣
 ↪include 0")
print("Practically significant:",False,",  CI include d_min")
```

```
Confidence Interval:[0.008104435728019967,0.05408517368626556]
Observed: 0.031094804707142765
Statistically significant: True ,  CI doesn't include 0
Practically significant: False ,  CI include d_min
```

[87]:
```
# Net Conversion
n=len(enrolls_exp)
d_min=0.0075
sum_clicks_cont=sum(clicks_cont[:n])
sum_clicks_exp=sum(clicks_exp[:n])
sum_payment_cont=sum(payment_cont[:n])
sum_payment_exp=sum(payment_exp[:n])
p_pool=(sum_payment_exp+sum_payment_cont)/(sum_clicks_exp+sum_clicks_cont)
SE_pool=math.sqrt(p_pool*(1-p_pool)*(1/sum_clicks_cont+1/sum_clicks_exp))
m=SE_pool*1.96
d_hat=sum_payment_exp/sum_clicks_exp-sum_payment_cont/sum_clicks_cont
print("Confidence Interval:[{},{}]".format(d_hat-m,d_hat+m))
print("Observed:",d_hat)
print ("Statistically significant:", d_hat+m<0 or d_hat-m>0 ,",  CI doesn't␣
 ↪include 0")
print("Practically significant:",False,",  CI include d_min")
```

```
Confidence Interval:[-0.011604624359891718,0.001857179010803383]
Observed: -0.0048737226745441675
Statistically significant: False ,  CI doesn't include 0
Practically significant: False ,  CI include d_min
```

Sign Tests For each of your evaluation metrics, do a sign test using the day-by-day data, and report the p-value of the sign test and whether the result is statistically significant.

[89]:
```
from scipy.stats import binom_test
# gross conversion
gc_exp=[i/j for i,j in zip(enrolls_exp,clicks_exp)]
gc_cont=[i/j for i,j in zip(enrolls_cont,clicks_cont)]
gc_diff=sum([i>j for i,j in zip(gc_exp,gc_cont)])
days=len(gc_exp)

# The prob of gross conversion of experiment group > gross conversion of control␣
 ↪group is 0.5
p_value=binom_test(gc_diff, n=days, p=0.5)
print("p-value:",p_value,", Statistically Significant:",p_value<0.05)
```

```
p-value: 0.0025994777679443364 , Statistically Significant: True
```

[90]:
```
# retention
rt_exp=[i/j for i,j in zip(payment_exp,enrolls_exp)]
rt_cont=[i/j for i,j in zip(payment_cont,enrolls_cont)]
rt_diff=sum([i>j for i,j in zip(rt_exp,rt_cont)])
days=len(rt_exp)
p_value=binom_test(rt_diff, n=days, p=0.5)
print("p-value:",p_value,", Statistically Significant:",p_value<0.05)
```

```
p-value: 0.6776394844055175 , Statistically Significant: False
```

[91]:
```
# net conversion
nc_exp=[i/j for i,j in zip(payment_exp,clicks_exp)]
nc_cont=[i/j for i,j in zip(payment_cont,clicks_cont)]
nc_diff=sum([i>j for i,j in zip(nc_exp,nc_cont)])
days=len(nc_exp)
p_value=binom_test(nc_diff, n=days, p=0.5)
print("p-value:",p_value,", Statistically Significant:",p_value<0.05)
```

```
p-value: 0.6776394844055175 , Statistically Significant: False
```

Summary State whether you used the Bonferroni correction, and explain why or why not. If there are any discrepancies between the effect size hypothesis tests and the sign tests, describe the discrepancy and why you think it arose.

I didn't use Bonferroni correction since it is too conservative and based on size tests, we could see most metrics nonsignificant. The reason why I think there is some discrepancies in Retention is because customers might pay their courses not day-by-day.

1.3 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, briefIy describe that experiment. If it is a judgment call, explain what factors would be relevant to the decision.

I wouldn't launch this experiment. It is clear that when it comes to payment, we could derive a conclusion that they will more likely pay for enrolled classes.

1.4 Follow-Up Experiment

Give a high-level description of the follow up experiment you would run, what your hypothesis would be, what metrics you would want to measure, what your unit of diversion would be, and your reasoning for these choices.

I would have follow-up experiment on why students don't pay for classes. It is because we don't provide good quality courses?

I would set up an experiment that we provide longer free trials to see more retention and net conversion.

The metric I will choose is retention and unit of diversion is user-id because we would like to know behaviors of customers.