

分析A/B测试结果

这个项目可以帮你确认你已经掌握了统计课程中涵盖的所有内容。希望这个项目尽可能地涵盖所有内容。祝你好运！

目录

- [简介](#)
- [I - 概率](#)
- [II - A/B 测试](#)
- [III - 回归](#)

简介

通常情况下，A/B 测试由数据分析师和数据科学家来完成。如果你在一些实践工作中遇到过这方面的问题，那学习起来就会更加游刃有余。

对于这个项目，你将要了解的是电子商务网站运行的 A/B 测试的结果。你的目标是通过这个 notebook 来帮助公司弄清楚他们是否应该使用新的页面，保留旧的页面，或者应该将测试时间延长，之后再做出决定。

使用该 notebook 的时候，请同步学习课堂内容，并回答与每个问题相关的对应测试题目。 每个课堂概念的标签对应每个题目。这样可以确保你在完成项目的过程中的方法正确，并且你最终提交的内容会更加符合标准，不必担心出现错误。最后检查的时候，请确保你的提交内容符合 [审阅标准](https://review.udacity.com/#!/projects/37e27304-ad47-4eb0-a1ab-8c12f60e43d0/rubric) (https://review.udacity.com/#!/projects/37e27304-ad47-4eb0-a1ab-8c12f60e43d0/rubric) 中的所有标准。

I - 概率

让我们先导入库，然后开始你的任务吧。

```
In [1]: import pandas as pd
import numpy as np
import random
import matplotlib.pyplot as plt
%matplotlib inline
#We are setting the seed to assure you get the same answers on quizzes as we set up
random.seed(42)
```

1. 现在，导入 `ab_data.csv` 数据，并将其存储在 `df` 中。使用你的 **dataframe** 来回答课堂测试 1 中的问题。

a. 导入数据集，并在这里查看前几行：

```
In [2]: df = pd.read_csv('ab-data.csv')
df.head()
```

Out[2]:

	user_id	timestamp	group	landing_page	converted
0	851104	2017-01-21 22:11:48.556739	control	old_page	0
1	804228	2017-01-12 08:01:45.159739	control	old_page	0
2	661590	2017-01-11 16:55:06.154213	treatment	new_page	0
3	853541	2017-01-08 18:28:03.143765	treatment	new_page	0
4	864975	2017-01-21 01:52:26.210827	control	old_page	1

b. 使用下面的单元格来查找数据集中的行数。

```
In [3]: df.shape
```

```
Out[3]: (294478, 5)
```

c. 数据集中独立用户的数量。

```
In [4]: df.user_id.nunique()
```

```
Out[4]: 290584
```

d. 用户转化的比例。

```
In [5]: df.converted.sum()/df.converted.count()
```

```
Out[5]: 0.11965919355605512
```

e. `new_page` 与 `treatment` 不一致的次数。

```
In [6]: df[((df['group'] == 'treatment') == (df['landing_page'] == 'new_page')) == False].shape[0]
```

```
Out[6]: 3893
```

f. 是否有任何行存在缺失值？

```
In [7]: df.duplicated().sum()
```

```
Out[7]: 0
```

2. 对于 `treatment` 不与 `new_page` 一致的行或 `control` 不与 `old_page` 一致的行，我们不能确定该行是否真正接收到了新的或旧的页面。我们应该如何处理这些行？在课堂中的 **测试 2** 中，给出你的答案。

a. 现在，使用测试题的答案创建一个符合测试规格要求的新数据集。将新 `dataframe` 存储在 `df2` 中。

```
In [8]: #去掉treatment不与new_page一致的行
index_1 = df[((df['group'] == 'treatment') == (df['landing_page'] == 'new_page')) == False].index.tolist()
df2 = df.drop(index_1)
```

```
In [9]: # 检查是否还有treatment不与new_page一致的行
df2[((df2['group'] == 'treatment') == (df2['landing_page'] == 'new_page')) == False].shape[0]
```

```
Out[9]: 0
```

3. 使用 `df2` 与下面的单元格来回答课堂中的 **测试3**。

a. `df2` 中有多少唯一的 `user_id`？

```
In [10]: df2.user_id.nunique()
```

```
Out[10]: 290584
```

b. df2 中有一个重复的 user_id 。它是什么？

```
In [11]: #得出去掉重复行, 并且不保留任意一行
df_a = df2.drop_duplicates(subset=['user_id'],keep = False)
#得出去掉重复行, 并且保留任第一行
df_b = df2.drop_duplicates(subset=['user_id'],keep = 'first')
#得出重复行
df_b.append(df_a).drop_duplicates(subset=['user_id'],keep = False)
```

Out[11]:

	user_id	timestamp	group	landing_page	converted
1899	773192	2017-01-09 05:37:58.781806	treatment	new_page	0

```
In [12]: df2.query('user_id == "773192"')
```

Out[12]:

	user_id	timestamp	group	landing_page	converted
1899	773192	2017-01-09 05:37:58.781806	treatment	new_page	0
2893	773192	2017-01-14 02:55:59.590927	treatment	new_page	0

d. 删除 一个 含有重复的 user_id 的行， 但需要确保你的 dataframe 为 df2。

```
In [13]: df2 = df2.drop_duplicates(subset=['user_id'], keep='first')
df2.shape
```

Out[13]: (290584, 5)

4. 在下面的单元格中， 使用 df2 来回答与课堂中的 测试 4 相关的测试题目。

a. 不管它们收到什么页面， 单个用户的转化率是多少？

```
In [14]: df2.converted.mean()
```

Out[14]: 0.11959708724499628

b. 假定一个用户处于 control 组中， 他的转化率是多少？

```
In [15]: df2.query('group == "control").converted.mean()
```

Out[15]: 0.1203863045004612

c. 假定一个用户处于 treatment 组中， 他的转化率是多少？

```
In [16]: df2.query('group == "treatment").converted.mean()
```

Out[16]: 0.11880806551510564

d. 一个用户收到新页面的概率是多少？

```
In [17]: df2.query('landing_page == "new_page").count()[0]/df2.count()[0]
```

Out[17]: 0.5000619442226688

e. 使用这个问题的前两部分的结果，给出你的建议：你是否认为有证据表明一个页面可以带来更多的转化？在下面写出你的答案。

答：由前两部分的结果可以知道，整个数据中处于control组的用户的平均转化率比处于treatment组的用户的转化率高0.12个百分点，但我认为目前没有证据表明某一个页面可以带来更多的转化率。

II - A/B 测试

请注意，由于与每个事件相关的时间戳，你可以在进行每次观察时连续运行假设检验。

然而，问题的难点在于，一个页面被认为比另一页页面的效果好得多的时候你就要停止检验吗？还是需要一定时间内持续发生？你需要将检验运行多长时间来决定哪个页面比另一个页面更好？

一般情况下，这些问题是A / B测试中最难的部分。如果你对下面提到的一些知识点比较生疏，请先回顾课程中的“描述统计学”部分的内容。

1. 现在，你要考虑的是，你需要根据提供的所有数据做出决定。如果你想假定旧的页面效果更好，除非新的页面在类型I错误率为5%的情况下才能证明效果更好，那么，你的零假设和备择假设是什么？你可以根据单词或旧页面与新页面的转化率 p_{old} 与 p_{new} 来陈述你的假设。

答：

$$H_0 : p_{new} - p_{old} \leq 0$$

$$H_1 : p_{new} - p_{old} > 0$$

2. 假定在零假设中，不管是新页面还是旧页面， p_{new} and p_{old} 都具有等于 **转化** 成功率的“真”成功率，也就是说， p_{new} 与 p_{old} 是相等的。此外，假设它们都等于 **ab_data.csv** 中的 **转化** 率，新旧页面都是如此。

每个页面的样本大小要与 **ab_data.csv** 中的页面大小相同。

执行两次页面之间 **转化** 差异的抽样分布，计算零假设中10000次迭代计算的估计值。

使用下面的单元格提供这个模拟的必要内容。如果现在还没有完整的意义，不要担心，你将通过下面的问题来解决这个问题。你可以通过做课堂中的 **测试 5** 来确认你掌握了这部分内容。

a. 在零假设中， p_{new} 的 **convert rate (转化率)** 是多少？

```
In [18]: p_new = df2.converted.mean()  
p_new
```

```
Out[18]: 0.11959708724499628
```

b. 在零假设中， p_{old} 的 **convert rate (转化率)** 是多少？

```
In [19]: p_old = df2.converted.mean()  
p_old
```

```
Out[19]: 0.11959708724499628
```

c. n_{new} 是多少?

```
In [20]: n_new = df2.query('landing_page == "new_page").count()[0]  
n_new
```

```
Out[20]: 145310
```

d. n_{old} 是多少?

```
In [21]: n_old = df2.query('landing_page == "old_page").count()[0]  
n_old
```

```
Out[21]: 145274
```

e. 在零假设中, 使用 p_{new} 转化率模拟 n_{new} 交易, 并将这些 n_{new} 1's 与 0's 存储在 **new_page_converted** 中。(提示: 可以使用 [numpy.random.choice](https://docs.scipy.org/doc/numpy/reference/generated/numpy.random.choice.html) (<https://docs.scipy.org/doc/numpy/reference/generated/numpy.random.choice.html>)).

```
In [22]: new_page_converted = np.random.choice(2, n_new, replace=True, p=[1-p_new, p_new])
```

f. 在零假设中, 使用 p_{old} 转化率模拟 n_{old} 交易, 并将这些 n_{old} 1's 与 0's 存储在 **old_page_converted** 中。

```
In [23]: old_page_converted = np.random.choice(2, n_old, replace=True, p=[1-p_old, p_old])
```

g. 在 (e) 与 (f) 中找到 $p_{new} - p_{old}$ 模拟值。

```
In [24]: new_page_converted.mean() - old_page_converted.mean()
```

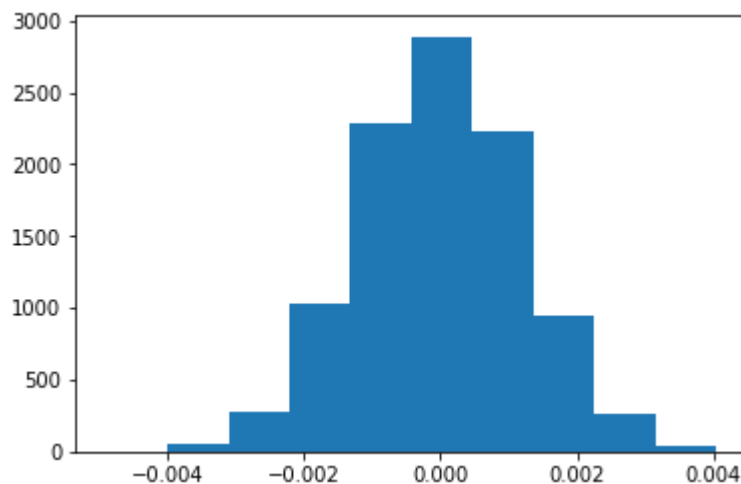
```
Out[24]: -9.145625285200687e-06
```

h. 使用 a. 到 g. 中的计算方法来模拟 10,000 个 $p_{new} - p_{old}$ 值, 并将这 10,000 个值存储在 **p_diffs** 中。

```
In [25]: p_diffs = []  
for _ in range(10000):  
    new_page_converted = np.random.choice(2, n_new, replace=True, p=[1-p_new, p_old])  
    old_page_converted = np.random.choice(2, n_old, replace=True, p=[1-p_old, p_old])  
    p_diffs.append(new_page_converted.mean() - old_page_converted.mean())
```

i. 绘制一个 **p_diffs** 直方图。这个直方图看起来像你所期望的吗? 通过回答课堂上的匹配问题, 确保你完全理解这里计算出的内容。

```
In [26]: plt.hist(p_diffs);
```

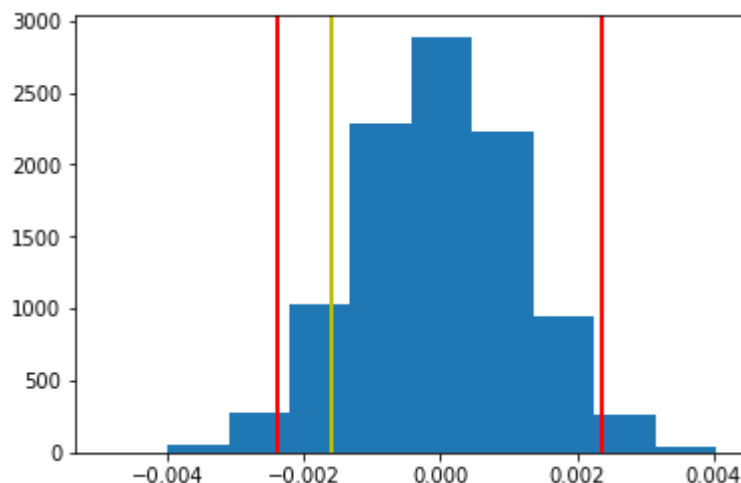


j. 在 **p_diffs** 列表的数值中，有多大比例大于 **ab_data.csv** 中观察到的实际差值？

```
In [27]: p_diff = df2.query('landing_page == "new_page").converted.mean() - df2.query('landing_page ==  
"old_page").converted.mean()  
p_diff
```

```
Out[27]: -0.0015782389853555567
```

```
In [28]: low, upper = np.percentile(p_diffs, 2.5), np.percentile(p_diffs, 97.5)  
plt.hist(p_diffs);  
plt.axvline(x=low, color='r', linewidth=2);  
plt.axvline(x=upper, color='r', linewidth=2);  
plt.axvline(x=p_diff, color='y', linewidth=2);
```



```
In [29]: (np.array(p_diffs) > p_diff).mean()
```

```
Out[29]: 0.9081
```

k. 用文字解释一下你刚才在 j. 中计算出来的结果。在科学研究中，这个值是什么？根据这个数值，新旧页面的转化率是否有区别呢？

答：这个j中计算出的结果即为假设检验中的P值，即当零假设为真的条件下，观察到样本统计量的概率，而题目前面给定错误率为5%,P足够大，所以我们根据样本数据不能够拒绝零假设，即新旧页面的转化率没有差别

l. 我们也可以使用一个内置程序（built-in）来实现类似的结果。尽管使用内置程序可能更易于编写代码，但上面的内容是对正确思考统计显著性至关重要的思想的一个预排。填写下面的内容来计算每个页面的转化次数，以及每个页面的访问人数。使用 `n_old` 与 `n_new` 分别引证与旧页面和新页面关联的行数。

```
In [30]: import statsmodels.api as sm
from scipy.stats import norm
convert_old = df2.query('landing_page == "old_page").converted.sum()
convert_new = df2.query('landing_page == "new_page").converted.sum()
n_old = df2.query('landing_page == "old_page").shape[0]
n_new = df2.query('landing_page == "new_page").shape[0]
```

```
C:\ProgramData\Anaconda3\lib\site-packages\statsmodels\compat\pandas.py:56: FutureWarning: The
e pandas.core.datetools module is deprecated and will be removed in a future version. Please
use the pandas.tseries module instead.
    from pandas.core import datetools
```

m. 现在使用 `stats.proportions_ztest` 来计算你的检验统计量与 p-值。这里

(http://knowledgetack.com/python/statsmodels/proportions_ztest/) 是使用内置程序的一个有用链接。

```
In [31]: #注意单尾
z_score, p_value = sm.stats.proportions_ztest([convert_new, convert_old], [n_new, n_old], alte
rnative='larger')
```

```
In [32]: z_score
```

```
Out[32]: -1.3109241984234394
```

```
In [33]: p_value
```

```
Out[33]: 0.9050583127590245
```

```
In [34]: #得出z_score的显著性
norm.cdf(z_score)
```

```
Out[34]: 0.09494168724097551
```

```
In [35]: #得出95%置信度临界值是多少
norm.ppf(0.05)
```

```
Out[35]: -1.6448536269514729
```

n. 根据上题算出的 z-score 和 p-value，我们认为新旧页面的转化率是否有区别？它们与 j. 与 k. 中的结果一致吗？

答：由计算结果，P值约为0.905，因此在95%的置信度下根据样本数据不可以拒绝零假设，即新旧页面转化率没有区别。并且这个P值与j-k中的P值大小基本相同，结论一致。

III - 回归分析法之一

1. 在最后一部分中，你会看到，你在之前的A / B测试中获得的结果也可以通过执行回归来获取。

a. 既然每行的值是转化或不转化，那么在这种情况下，我们应该执行哪种类型的回归？

答：在此处应使用分类数据的线性回归，设置虚拟变量（0,1），1代表转化0代表不转化。

b. 目标是使用 **statsmodels** 来拟合你在 a. 中指定的回归模型，以查看用户收到的不同页面是否存在显著的转化差异。但是，首先，你需要为这个截距创建一个列（原文：column），并为每个用户收到的页面创建一个虚拟变量列。添加一个 **截距** 列，一个 **ab_page** 列，当用户接收 **treatment** 时为1， **control** 时为0。

```
In [36]: df2_new = df2
#添加ab_page列，当用户接收 treatment 时为1， control 时为0
df2_new['ab_page'] = pd.get_dummies(df['group'])['treatment']
#添加截距列
df2_new['intercept'] = 1
df2_new.head()
```

Out[36]:

	user_id	timestamp	group	landing_page	converted	ab_page	intercept
0	851104	2017-01-21 22:11:48.556739	control	old_page	0	0	1
1	804228	2017-01-12 08:01:45.159739	control	old_page	0	0	1
2	661590	2017-01-11 16:55:06.154213	treatment	new_page	0	1	1
3	853541	2017-01-08 18:28:03.143765	treatment	new_page	0	1	1
4	864975	2017-01-21 01:52:26.210827	control	old_page	1	0	1

c. 使用 **statsmodels** 导入你的回归模型。 实例化该模型，并使用你在 b. 中创建的2个列来拟合该模型，用来预测一个用户是否会发生转化。

```
In [37]: log_mod = sm.Logit(df2_new['converted'], df2_new[['intercept', 'ab_page']])
results = log_mod.fit()

Optimization terminated successfully.
Current function value: 0.366118
Iterations 6
```

d. 请在下方提供你的模型摘要，并根据需要使用它来回答下面的问题。

```
In [38]: results.summary2()
```

Out[38]:

Model:	Logit	No. Iterations:	6.0000
Dependent Variable:	converted	Pseudo R-squared:	0.000
Date:	2018-05-29 18:10	AIC:	212780.3502
No. Observations:	290584	BIC:	212801.5095
Df Model:	1	Log-Likelihood:	-1.0639e+05
Df Residuals:	290582	LL-Null:	-1.0639e+05
Converged:	1.0000	Scale:	1.0000

	Coef.	Std.Err.	z	P> z	[0.025	0.975]
intercept	-1.9888	0.0081	-246.6690	0.0000	-2.0046	-1.9730
ab_page	-0.0150	0.0114	-1.3109	0.1899	-0.0374	0.0074

e. 与 `ab_page` 关联的 p-值是多少？为什么它与你在 II 中发现的结果不同？

提示: 与你的回归模型相关的零假设与备择假设分别是什么？它们如何与 **Part II** 中的零假设和备择假设做比较？

答：由模型摘要，与`ab_page`关联的p值为0.190，与partII中的结果不同。
原因是在此处的零假设和II中的零假设和备择假设不同，II中的零假设为：

$$H_0 : p_{new} - p_{old} \leq 0$$

而此处回归模型的零假设为：

$$H_0 : p_{new} - p_{old} = 0$$

因此两处计算的P值不同。

f. 现在，你一直在考虑其他可能影响用户是否发生转化的因素。讨论为什么考虑将其他因素添加到回归模型中是一个不错的主意。在回归模型中添加附加项有什么弊端吗？

答：当将其他因素添加到回归模型中构建多重线性回归模型时，会有如下一些潜在问题：

- 1. 因变量-自变量关系的非线性
- 2. 误差项的相关性
- 3. 非恒定方差和正态分布误差
- 4. 异常值/高杠杆点
- 5. 共线性

g. 现在，除了测试不同页面的转化率是否会发生变化之外，还要根据用户居住的国家或地区添加一个 `effect` 项。你需要导入 `countries.csv` 数据集，并将数据集合并到适当的行上。 [这里 \(https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.join.html\)](https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.join.html) 是链接表格的文档。

这个国家项对转化有影响吗？不要忘记为这些国家的列创建虚拟变量—— **提示: 你将需要为这三个虚拟变量增加两列。** 提供统计输出，并书面回答这个问题。

```
In [39]: df3 = pd.read_csv('countries.csv')
df3.head()
```

Out[39]:

	user_id	country
0	834778	UK
1	928468	US
2	822059	UK
3	711597	UK
4	710616	UK

In [40]:

#按照user_id合并数据集
df3_new = pd.merge(df3, df2_new, on='user_id')
df3_new.head()

Out[40]:

	user_id	country	timestamp	group	landing_page	converted	ab_page	intercept
0	834778	UK	2017-01-14 23:08:43.304998	control	old_page	0	0	1
1	928468	US	2017-01-23 14:44:16.387854	treatment	new_page	0	1	1
2	822059	UK	2017-01-16 14:04:14.719771	treatment	new_page	1	1	1
3	711597	UK	2017-01-22 03:14:24.763511	control	old_page	0	0	1
4	710616	UK	2017-01-16 13:14:44.000513	treatment	new_page	0	1	1

In [41]:

#添加UK列和CA列
df3_new[['UK', 'CA']] = pd.get_dummies(df3_new['country'])[['UK', 'CA']]
df3_new.head()

Out[41]:

	user_id	country	timestamp	group	landing_page	converted	ab_page	intercept	UK
0	834778	UK	2017-01-14 23:08:43.304998	control	old_page	0	0	1	1
1	928468	US	2017-01-23 14:44:16.387854	treatment	new_page	0	1	1	0
2	822059	UK	2017-01-16 14:04:14.719771	treatment	new_page	1	1	1	1
3	711597	UK	2017-01-22 03:14:24.763511	control	old_page	0	0	1	1
4	710616	UK	2017-01-16 13:14:44.000513	treatment	new_page	0	1	1	1

```
In [42]: logit_mod = sm.Logit(df3_new['converted'], df3_new[['intercept', 'UK', 'CA']])
results = logit_mod.fit()
results.summary2()
```

Optimization terminated successfully.
Current function value: 0.366116
Iterations 6

Out[42]:

Model:	Logit	No. Iterations:	6.0000
Dependent Variable:	converted	Pseudo R-squared:	0.000
Date:	2018-05-29 18:10	AIC:	212780.8333
No. Observations:	290584	BIC:	212812.5723
Df Model:	2	Log-Likelihood:	-1.0639e+05
Df Residuals:	290581	LL-Null:	-1.0639e+05
Converged:	1.0000	Scale:	1.0000

	Coef.	Std.Err.	z	P> z	[0.025	0.975]
intercept	-1.9967	0.0068	-292.3145	0.0000	-2.0101	-1.9833
UK	0.0099	0.0133	0.7458	0.4558	-0.0161	0.0360
CA	-0.0408	0.0269	-1.5178	0.1291	-0.0935	0.0119

由以上摘要，根据P值的大小可以得出结论：国家这一项对转化率的影响不具有统计显著性。

h. 虽然你已经查看了国家与页面在转化率上的个体性因素，但现在我们要查看页面与国家/地区之间的相互作用，测试其是否会对转化产生重大影响。创建必要的附加列，并拟合一个新的模型。

提供你的摘要结果，以及根据结果得出的结论。

提示： 页面与国家/地区的相互作用

```
df3['new_CA'] = df3['new_page'] * df3['CA']
df3['new_UK'] = df3['new_page'] * df3['UK']
```

```
In [43]: #添加new_page列, 当landing_page为new_page时值为1
df3_new['new_page'] = pd.get_dummies(df['landing_page'])['new_page']
#添加页面与国家/地区的相互作用列
df3_new['new_CA'] = df3_new['new_page'] * df3_new['CA']
df3_new['new_UK'] = df3_new['new_page'] * df3_new['UK']
df3_new.head()
```

Out[43]:

	user_id	country	timestamp	group	landing_page	converted	ab_page	intercept	UK
0	834778	UK	2017-01-14 23:08:43.304998	control	old_page	0	0	1	1
1	928468	US	2017-01-23 14:44:16.387854	treatment	new_page	0	1	1	0
2	822059	UK	2017-01-16 14:04:14.719771	treatment	new_page	1	1	1	1
3	711597	UK	2017-01-22 03:14:24.763511	control	old_page	0	0	1	1
4	710616	UK	2017-01-16 13:14:44.000513	treatment	new_page	0	1	1	1

```
In [47]: log_mod = sm.Logit(df3_new['converted'], df3_new[['intercept', 'new_UK', 'new_CA', 'new_page', 'UK', 'CA']])
results = log_mod.fit()
results.summary2()
```

Optimization terminated successfully.
Current function value: 0.366113
Iterations 6

Out[47]:

Model:	Logit	No. Iterations:	6.0000
Dependent Variable:	converted	Pseudo R-squared:	0.000
Date:	2018-05-29 19:01	AIC:	212785.1159
No. Observations:	290584	BIC:	212848.5938
Df Model:	5	Log-Likelihood:	-1.0639e+05
Df Residuals:	290578	LL-Null:	-1.0639e+05
Converged:	1.0000	Scale:	1.0000

	Coef.	Std.Err.	z	P> z	[0.025	0.975]
intercept	-2.0014	0.0097	-206.6510	0.0000	-2.0204	-1.9824
new_UK	-0.0176	0.0266	-0.6600	0.5093	-0.0697	0.0346
new_CA	0.0458	0.0538	0.8511	0.3947	-0.0596	0.1512
new_page	0.0093	0.0137	0.6821	0.4951	-0.0175	0.0361
UK	0.0187	0.0188	0.9959	0.3193	-0.0181	0.0554
CA	-0.0641	0.0385	-1.6650	0.0959	-0.1395	0.0114

由以上模型摘要，根据P值打小，在95%的置信度下，我们不能认为页面与国家/地区之间的相互作用对转化有重大影响。

总结

祝贺你顺利完成了该项目！

收集提交材料

你对你的 notebook 的状态感到满意后，应该将其保存为便于其他人阅读并查看的格式。你可以使用 **File -> Download as -> HTML (.html)** 菜单将你的 Notebook 保存为一个 .html 文件。如果你在本地操作，并发现了一个 "找不到模块名称 (No module name) " 错误，请打开终端并尝试使用 `pip install <module_name>` 安装缺少的模块（注：在模块名称中，不要包含 "<" or ">" 或任何带有期限的词汇）。

你需要同时提交原始 Notebook 和 Notebook 的HTML或PDF副本以供审阅。你不需要在提交中附带任何数据文件。如果你参考了其他网站、书籍和其他资源来帮助你解决项目中的任务，请确保记录在案。建议你在 Notebook报告末尾的 Markdown 单元格中添加 “Resources” 部分，或者可以附带一个记录你的参考资源的 `readme.txt` 文件。

提交项目

准备就绪后，点击“提交项目”按钮进入项目提交页面。你可以将文件以 .zip压缩文件提交，也可以链接到包含项目文件的 GitHub 存储库。如果你使用的是 GitHub，请注意，你提交的内容将是提交时的链接库的一个快照。建议你将每个项目保存在一个单独的存储库中，以避免出现混淆：如果审阅专家获取多个文件夹来代表多个项目，则他可能不明确要评估哪个项目。

我们需要一周的时间对项目进行评分，但在大多数情况下，速度要快得多。当你的提交被审阅后，你将会收到一封电子邮件。如果你在提交项目时遇到任何问题，或者想要查看提交状态，请发送电子邮件至 support@youdaxue.com。同时，你可以继续学习下一个单元的课程。