Localized Campaign A/B Testing

Alireza Kazemi

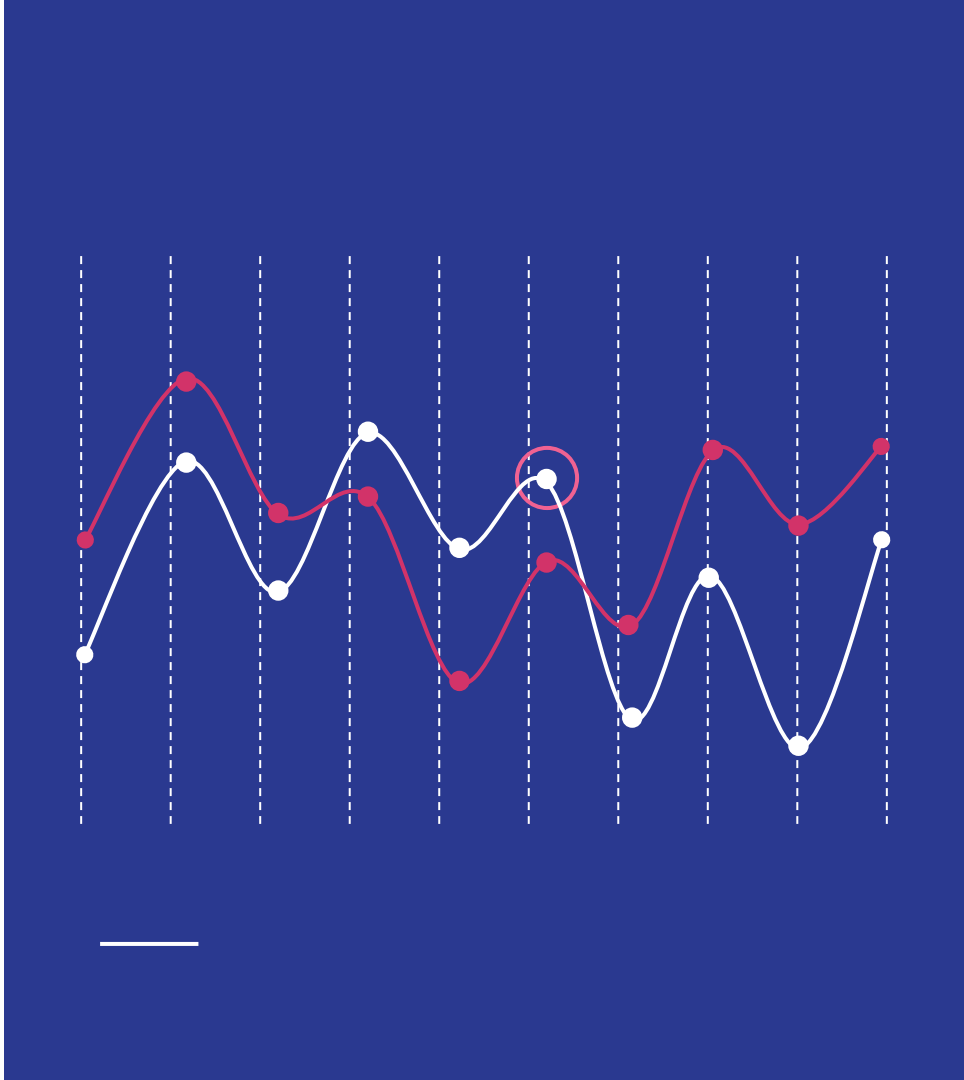


Table Schema

The test has launched in 17 cities for five days, for each customer we have a single record. The figure below shows us the information about every customer's action. (advertising channel for near half of data is missed)

	customer_id	date	source	device	browser_language	ads_channel	browser	conversion	test	sex	age	city
0	315281	12/3/2019	Direct	Web	EN	NaN	Edge	1	0	M	32	Calgary
1	497851	12/4/2019	Ads	Web	EN	Google	Edge	0	1	M	21	Toronto
2	848402	12/4/2019	Ads	Web	EN	Facebook	Chrome	0	0	M	34	Calgary
3	290051	12/3/2019	Ads	Mobile	Other	Facebook	Android_App	0	1	F	22	Toronto
4	548435	11/30/2019	Ads	Web	EN	Google	FireFox	0	1	M	19	Toronto
5	540675	12/3/2019	Direct	Mobile	EN	NaN	Android_App	0	1	F	22	Ottawa
6	863394	12/4/2019	SEO	Mobile	Other	NaN	Android_App	0	0	M	35	Toronto
7	527287	12/3/2019	Direct	Web	FR	NaN	Chrome	0	0	M	22	Calgary
8	261625	12/4/2019	Direct	Mobile	EN	NaN	Android_App	0	1	M	31	Kitchener
9	10427	12/4/2019	Ads	Mobile	EN	Facebook	Android_App	0	0	F	33	Toronto

Test Results

The overall result of our test shows the conversion rate has absolutely **decreased by 1.2%** from 5.5% (Control) with a **99% significance** and it seems the new campaign was not successful.

OLS Regression Results

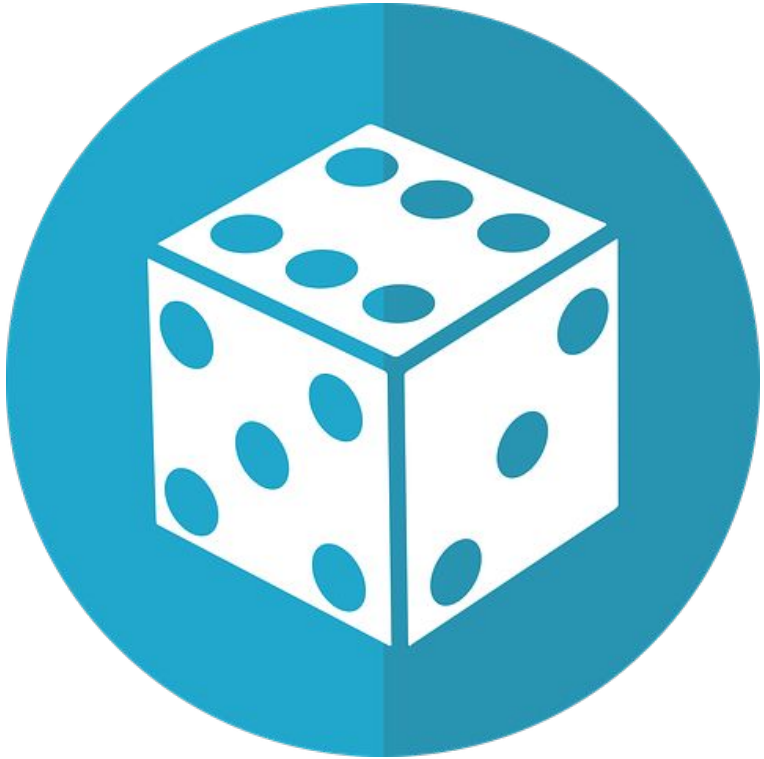
Dep. Variable:	conversion	R-squared:	0.001
Model:	OLS	Adj. R-squared:	0.001
Method:	Least Squares	F-statistic:	331.0
Date:	Thu, 07 Jul 2022	Prob (F-statistic):	6.09e-74
Time:	10:43:04	Log-Likelihood:	49421.
No. Observations:	452867	AIC:	-9.884e+04
Df Residuals:	452865	BIC:	-9.882e+04
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	0.0552	0.000	123.788	0.000	0.054	0.056
test	-0.0117	0.001	-18.194	0.000	-0.013	-0.010

Omnibus:	370070.056	Durbin-Watson:	1.999
Prob(Omnibus):	0.000	Jarque-Bera (JB):	5661485.306
Skew:	4.146	Prob(JB):	0.00
Kurtosis:	18.207	Cond. No.	2.57

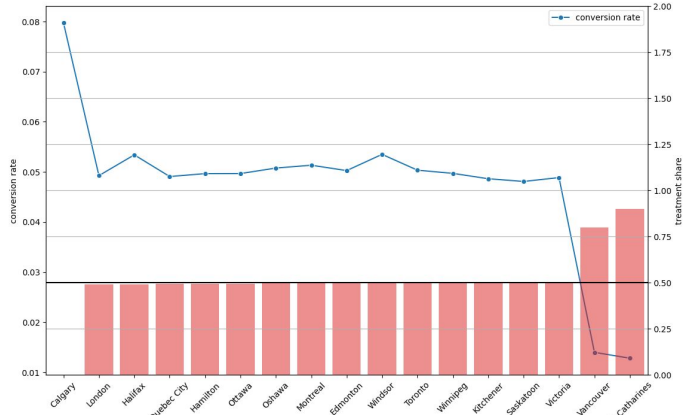
Randomized Experiment

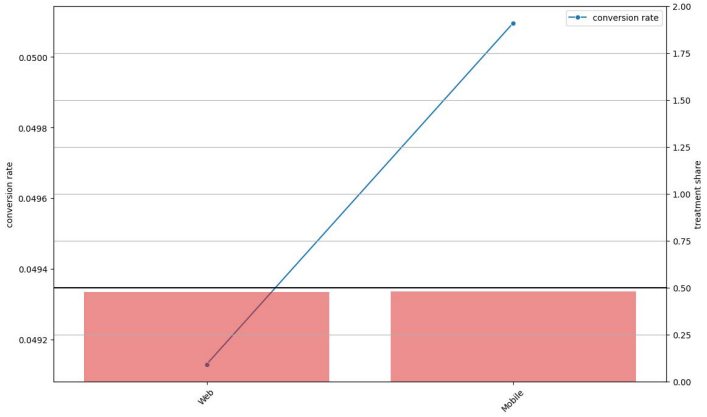
Now, our question is either our experiment ran fairly or not. So, we can check the results in different segments such as **device, date, language and city**.

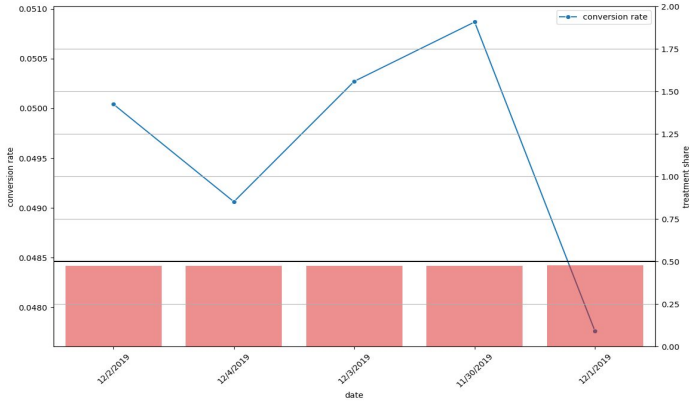


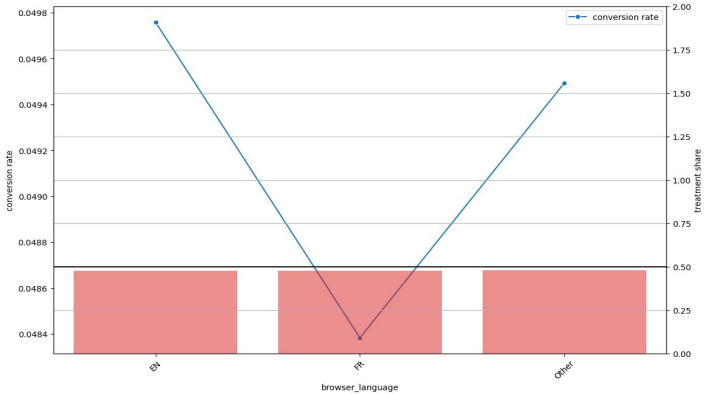
Segmentation

As we can see in three segments (date, device, and language) the experiment was almost fair. The share of test groups in these segments is nearly equal, but, **the test has a bias on cities**.



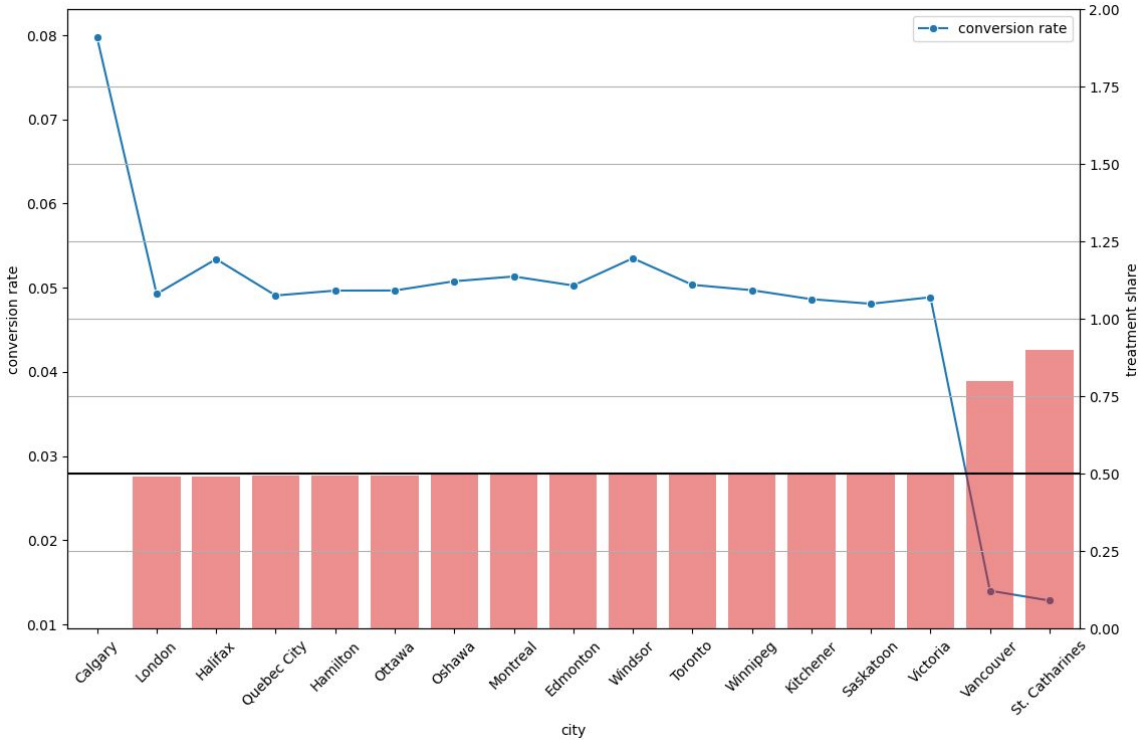






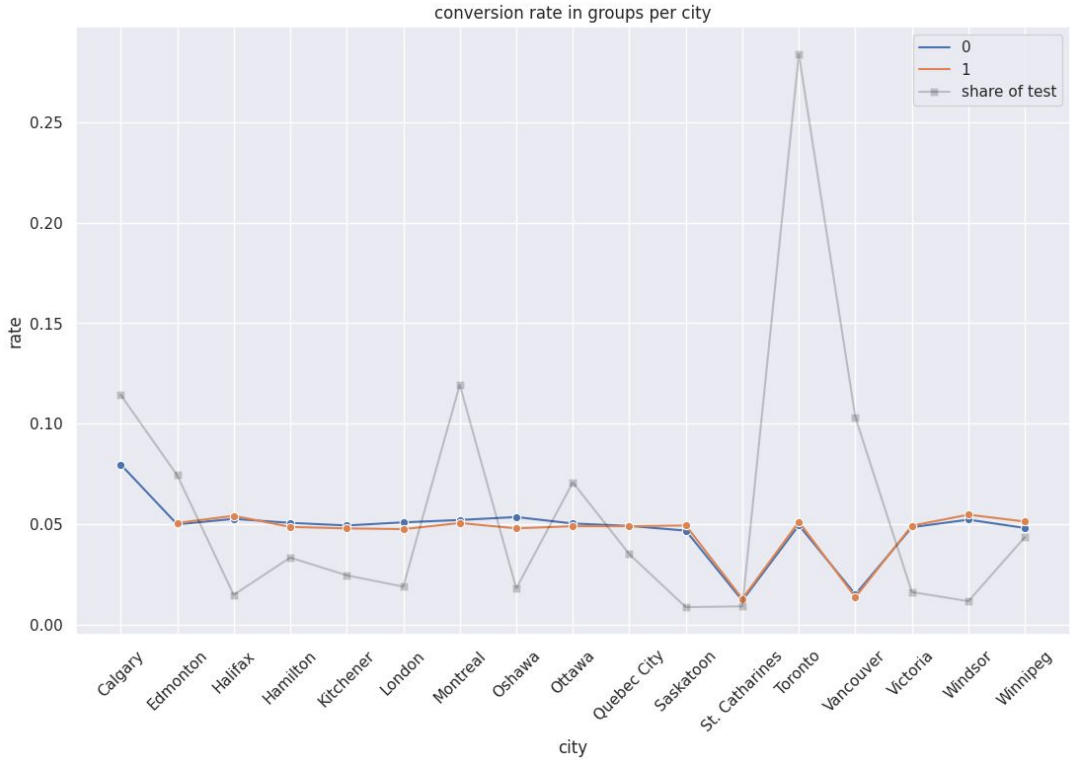
Bias on City

Here, we don't have test sample for **Calgary** which is its conversion rate is highly more than average. And, the share of test group is high in **Vancouver** and **St. Catharines** with a low average conversion rate.



Bias on City

This figure shows the share of users in different cities and their conversion rates for treatment and control. The share of Calgary users (whose conversion is 8%) is about 12% causes an unfair increase in the control conversion rate.



Fair Results

Now, if we estimate the conversion rate by the city and test group, the results show no significant difference in test groups. (p-value = 0.84)

OLS Regression Results

Dep. Variable:	conversion	R-squared:	0.005
Model:	OLS	Adj. R-squared:	0.005
Method:	Least Squares	F-statistic:	141.0
Date:	Thu, 07 Jul 2022	Prob (F-statistic):	0.00
Time:	11:12:19	Log-Likelihood:	50451.
No. Observations:	452867	AIC:	-1.009e+05
Df Residuals:	452849	BIC:	-1.007e+05
Df Model:	17		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	0.0797	0.001	83.803	0.000	0.078	0.082
city[T.Edmonton]	-0.0295	0.002	-18.987	0.000	-0.033	-0.026
city[T.Halifax]	-0.0264	0.003	-9.334	0.000	-0.032	-0.021
city[T.Hamilton]	-0.0301	0.002	-14.840	0.000	-0.034	-0.026
city[T.Kitchener]	-0.0312	0.002	-13.610	0.000	-0.036	-0.027
city[T.London]	-0.0305	0.003	-11.984	0.000	-0.036	-0.026
city[T.Montreal]	-0.0285	0.001	-20.678	0.000	-0.031	-0.026
city[T.Oshawa]	-0.0290	0.003	-11.165	0.000	-0.034	-0.024
city[T.Ottawa]	-0.0301	0.002	-19.098	0.000	-0.033	-0.027
city[T.Quebec City]	-0.0307	0.002	-15.411	0.000	-0.035	-0.027
city[T.Saskatoon]	-0.0317	0.004	-8.830	0.000	-0.039	-0.025
city[T.St. Catharines]	-0.0670	0.004	-18.855	0.000	-0.074	-0.060
city[T.Toronto]	-0.0294	0.001	-24.955	0.000	-0.032	-0.027
city[T.Vancouver]	-0.0658	0.001	-44.170	0.000	-0.069	-0.063
city[T.Victoria]	-0.0309	0.003	-11.363	0.000	-0.036	-0.026
city[T.Windsor]	-0.0263	0.003	-8.377	0.000	-0.032	-0.020
city[T.Winnipeg]	-0.0301	0.002	-16.311	0.000	-0.034	-0.026
test	0.0001	0.001	0.201	0.840	-0.001	0.002

Omnibus:	368028.485	Durbin-Watson:	2.000
Prob(Omnibus):	0.000	Jarque-Bera (JB):	5560415.924
Skew:	4.117	Prob(JB):	0.00
Kurtosis:	18.062	Cond. No.	17.7

Bias Detector

We want to automatically detect biases in other tests, so we can first calculate the overall result and compare it with results which conditioned by different segments of test. Here, we need a definition ACE (Average Causal Effect) which removes the bias effect.

Randomization Solves the Selection Problem

Recall the selection bias formula:

$$\begin{aligned}
\tilde{\tau} &= \mathbb{E}[Y_i|D_i=1] - \mathbb{E}[Y_i|D_i=0] \quad \text{(observed difference in means)} \\
&= \mathbb{E}[Y_{1i}|D_i=1] - \mathbb{E}[Y_{0i}|D_i=0] \\
&= \underbrace{\mathbb{E}[Y_{1i} - Y_{0i}|D_i=1]}_{\tau_{ATT}} + \underbrace{\mathbb{E}[Y_{0i}|D_i=1] - \mathbb{E}[Y_{0i}|D_i=0]}_{\text{Bias}}
\end{aligned}$$

How can we eliminate the bias term?

Random assignment of D_i will make the treated and untreated units identical on average, such that

$$\mathbb{E}[Y_{0i} \mid D_i=1] = \mathbb{E}[Y_{0i} \mid D_i=0]$$

This implies Bias $= 0$.

Bias Detector

This function takes the segments that may create bias and remove the groups without fair assignment (for example, in our task Calgary will be removed). Then, it calculates each test group's conversion rate considering a group share in the whole test and compares it with the overall score. If at least a score was different, it will return False.

```
def score_calculator (df_data,column_list,fair_thresh):
    first_result = df_customers.dropna(subset=['test','conversion']).groupby(['test']).mean().sort_values(by='test')['conversion']
    raw_score = np.round(100*(first_result[1]-first_result[0]),1)

    socre_map = {"Cateogory":[],"Test Score":[]}
    for column in column_list:

      df_data.dropna(subset=[column],inplace=True)
      diff_test = df_data.groupby(['test',column],as_index=False).mean()[[column,'test','conversion']]
      column_share = df_data.groupby([column],as_index=False).count()[[column,'customer_id']]
      test_share = df_data.groupby([column],as_index=False).mean()[[column,'test']]
      column_share['share'] = column_share['customer_id']/np.sum(column_share['customer_id'])

      do_calculus = diff_test.merge(column_share,on=column).merge(test_share,on=column).rename(columns={'test_y': 'test_share','test_x':'test'})

      final = do_calculus.where(
          abs(do_calculus['test_share']-0.5)<=fair_thresh
      ).dropna().head(20)

      final['weight'] = final['conversion']*final['share']
      treatment_df = final.where(final['test']==1)
      treatment_score = np.round(100*np.sum(treatment_df['weight'])/np.sum(treatment_df['share']),2)
      control_df = final.where(final['test']==0)
      control_score = np.round(100*np.sum(control_df['weight'])/np.sum(control_df['share']),2)
      test_score = np.round(treatment_score - control_score,1)
      socre_map["Cateogory"].append(column)
      socre_map["Test Score"].append(test_score)

    result = pd.DataFrame(socre_map)
    avg = np.mean(result['Test Score'])
    result['NotFair'] = result['Test Score'].apply(lambda x:int(abs(x/raw_score - 1)>=fair_thresh))
    if np.sum(result['NotFair'])>=1):
      print(result)
      return False

    else:
      print(result)
      return True
```

Example

For example, for this test the overall score of test was -1.2%. Average causal effect was same for other segments such as browser, device, date, sex, source, language, and age, but, ACE for city was about -0.1%, so, for some reasons our random assignment was not fair.

```
[ ]  score_calculator(df_customers,['city','browser','device','sex','date','browser_language','source','age'],0.1)
```

```
           Cateogory  Test Score  NotFair
0               city        -0.1        1
1            browser        -1.2        0
2             device        -1.2        0
3                sex        -1.2        0
4               date        -1.2        0
5  browser_language        -1.2        0
6             source        -1.2        0
7                age        -1.1        0
False
```

The End