Section II: Checking Conversion Rates

In [97]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm
```

In [98]:

```
df = pd.read_csv("conversion_rates.csv")
```

In [99]:

```
df.head()
```

Out[99]:

	date	male	age	has_insurance	came_from	reached_end
0	2018-09-03	1	32	0	Insurance Site B	1
1	2018-09-07	1	35	0	Insurance Site A	0
2	2018-09-05	1	34	0	Insurance Site C	1
3	2018-09-03	1	32	0	Insurance Site C	1
4	2018-09-05	1	31	0	Google Search	0

In [100]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 280 entries, 0 to 279
Data columns (total 6 columns):
date                 280 non-null object
male                 280 non-null int64
age                  280 non-null int64
has_insurance        280 non-null int64
came_from            280 non-null object
reached_end          280 non-null int64
dtypes: int64(4), object(2)
memory usage: 13.2+ KB
```

In [102]:

```
# Sorting data in ascending order
df = df.sort_values(['date'])
```

In [103]:

```
df.head()
```

Out[103]:

	date	male	age	has_insurance	came_from	reached_end
200	2018-09-02	1	35	1	Insurance Site C	1
199	2018-09-02	1	29	1	Google Search	0
198	2018-09-02	0	32	1	Google Search	0
197	2018-09-02	1	32	0	Google Search	1
104	2018-09-02	1	36	1	Insurance Site C	0

In [104]:

```
#dataframe after 2018-09-05
df2 = df[(df['date'] > "2018-09-04")]
#dataframe before 2018-09-05
df1 = df[(df['date'] <= "2018-09-04")]
```

In [105]:

```
#list of conversion_rate before 2018-09-05
df1_num_con = list(df1.groupby("reached_end")['age'].count())
#list of conversion_rate after 2018-09-05
df2_num_con = list(df2.groupby("reached_end")['age'].count())
# mean of conversion rate before 2018-09-05
df1_mean = df1['reached_end'].mean()
# mean of conversion rate after 2018-09-05
df2_mean = df2['reached_end'].mean()
# standard deviation of conversion rate after 2018-09-05
df1_std = df1['reached_end'].std()
# standard deviation of conversion rate before 2018-09-05
df2_std = df2['reached_end'].std()
```

In [106]:

```
df1_mean, df2_mean
```

Out[106]:

```
(0.3383458646616541, 0.5782312925170068)
```

As we can already see from the mean of both the samples, sample 2 (after product change date) has a higher mean of conversion than sample 1 (before product change)

We should do a 2 sample Hypothesis test to check for the same

Ho = Product change did not improve the conversion rate
Ha = Product change did improve the conversion rate
alpha (significance level) = 0.05

Now let's do a 2 sample Z test on our samples
And the formula is:

$$z = \frac{(\overline{X}_1 - \overline{X}_2) - (\mu_1 - \mu_2)}{\sqrt{\sigma^2_{\overline{X}_1} + \sigma^2_{\overline{X}_2}}} = \frac{(\overline{X}_1 - \overline{X}_2) - (\mu_1 - \mu_2)}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}$$

In [139]:

```
# Function for calculating the Z and p-value
def two_sam_Z(X1, X2, mudiff, sd1, sd2, n1, n2):

    pooledSE = np.sqrt(sd1**2/n1 + sd2**2/n2)
    z = ((X1 - X2) - mudiff)/pooledSE
    pval = (1 - norm.cdf(abs(z)))
    return round(z, 3), round(pval, 5)
```

In [140]:

```
two_sam_Z(df1_mean, df2_mean, 0, df1_std, df2_std, sum(df1_num_con), sum
(df2_num_con))
```

Out[140]:

```
(-4.134, 2e-05)
```

As we can see p-value is 0.00002, which is << our significance level so we reject the null hypothesis.
',' There is a 0.002% chance that the product change did not improve the conversion rate.
So we should go ahead with the product improvement!

In [ ]:

```

```