AB Testing @ Photobox

Nathalie Skrzypek

Introduction & Agenda

photoboxgroup

Photobox - largest personalised photo business in Europe

The Goal - Optimise our website (happier customers, higher margins)

The Problem - While we think we know best practices, the opinion of the masses is what counts.

The Solution - AB testing

Agenda

- Intro to AB testing
- Statistics behind AB testing
- Interpretation of Results

Logistics of AB Testing

photoboxgroup

- Frequentist AB Testing
- Version A: Control
- Version B: Contender
- Data Scientist calculates length of test to be significant (based on effect_size, type I/II error, test type)
- 50:50 traffic split using Optimizely
- Calculate P-value: The probability under the null hypothesis of obtaining a result equal to or more extreme than what was observed

```
from statsmodels.stats import proportion, power
es = proportion.proportion_effectsize(cr_1, cr_0, method='normal')
power.tt_ind_solve_power(effect_size=es,
                         nobs1=None,
                         alpha=alpha,
                         power=power,
                         ratio=1,
                         alternative='two-sided')
```

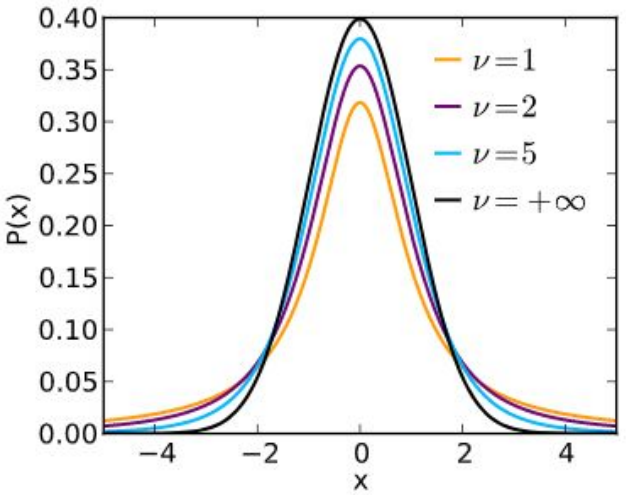
Statistics behind AB testing - Part I

photoboxgroup

Independent samples (2 tailed) Student's t-test

- Used for **continuous input** variables (e.g. margin at Photobox)
- Statistical hypothesis test based on the Student's t-distribution.
- Compares the sample means of 2 groups

```
def two_sided_t_test(df, title, margin, margin_name):
    df_0 = df[(df.variant_id==0)]
    df_1 = df[(df.variant_id==1)]
    t, p = stats.ttest_ind(df_0[margin].astype(float),
                                            df_1[margin].astype(float),
                                            equal_var=False)
    output = [title, round(t,5), round(p,5), round((1-p)*100, 2),
              round(df_0[margin].mean(),2),round(df_1[margin].mean(),2),
              round(df_0[margin].mean(),2) - round(df_1[margin].mean(),2), df_0.shape[0], df_1.shape[0]]
    print(significant(p, title),
          '\nTest Statistic:', round(t,5),
          '\np-value:', round(p,5),
          '\nConfidence:', round((1-p)*100, 2),
          '%\n', margin_name + ' Per User (Control): £', round(df_0[margin].mean(),2),
          '\n', margin_name + ' Per User (Contender): £', round(df_1[margin].mean(),2),
          '\nMargin Difference:', round((df_0[margin].mean() - df_1[margin].mean()),2),
          '\nSize (Control):', df_0.shape[0],
          '\nSize(Contender):', df_1.shape[0],
          '\n--------------------------\n')
    return pd.DataFrame([output])
```



Statistics behind AB testing - Part II

photoboxgroup

z-score

- For **binomial variables** (0 / 1, e.g. conversion rate at Photobox)
- Deviation from mean in units of standard deviations
- Like signal-to-noise: (X - m) / SE
- Sample mean is proportion of success

```
def cr_se(df):
    cr = float(sum(df.customers) / sum(df.users))
    se = float(np.sqrt(cr*(1-cr) / sum(df.users)))
    return cr*100, se*100

def z_score_p_value(df, title):
    df_0 = df[df.variant_id==0]
    df_1 = df[df.variant_id==1]

    cr_0, se_0 = cr_se(df_0)
    cr_1, se_1 = cr_se(df_1)

    z_score = (cr_0 - cr_1) / (np.sqrt((se_0)**2 + (se_1)**2))
    p_values = stats.norm.sf(abs(z_score))*2
    delta = ((cr_1 - cr_0) / cr_0)*100
    output = [title, round(cr_0,5), round(se_0,5), round(cr_1,5), round(se_1,5),
              df_0.shape[0], df_1.shape[0], round(z_score,5),
              round(p_values,5), round((1-p_values)*100,2), round(delta, 5)]
    print(significant(p_values, title),
          '\ncr_0:', round(cr_0,5),
          '\nse_0:', round(se_0,5),
          '\ncr_1:', round(cr_1,5),
          '\nse_1:', round(se_1,5),
          '\nSize (Control):', df_0.shape[0],
          '\nSize (Contender):', df_1.shape[0],
          '\nz_score:', round(z_score,5),
          '\np_values:', round(p_values,5),
          '\nsignificance:', round((1-p_values)*100,2),
          '\ndelta:', round(delta, 5),'%\n',
          '--------------------------\n')
    return pd.DataFrame([output])
```

Results

photoboxgroup

Photobox uses P <= 0.1 (False Positive Rate of 10%) as our Significance cut-off

- If P <=0.1 and positive (aka better margin / conversion rate for B) then 🎉🎉🎉 and implement B

- If P <= 0.1 and negative (aka worse margin / conversion rate for B) then 😱😱😱 and revert to A

- If P >= 0.1 then implement B anyways as it had no impact on margin / CR and B is more modern