1.数据来源

kaggle (https://www.kaggle.com/yufengsui/mobile-games-ab-testing)
(https://www.kaggle.com/yufengsui/mobile-games-ab-testing (https://www.kaggle.com/yufengsui/mobile-games-ab-testing))
该数据是一个手游的A/Btest结果

字段名称

- userid：用户id
- version：版本区别 **A**：gate_30 **B**：gate_40
- sum_gamerounds：安装后14天内，玩家玩的游戏回合数。
- retention_1：次日留存率
- retention_7：7日留存率

2.分析目的

- 根据假设检验的统计学原理确认哪个项目表现更好

3.数据导入与清洗

- 空值的检查与处理
- 重复值的检查与处理
- 异常值的检查与处理
- 数据类型的检查与调节

In [1]:

```
#首先导入必需的第三方包
import numpy as np
import pandas as pd
import statsmodels.stats.proportion as ssp
```

In [2]:

```
#读取数据
data = pd.read_csv('cookie_cats.csv')
data.head()
```

Out[2]:

	userid	version	sum_gamerounds	retention_1	retention_7
0	116	gate_30	3	False	False
1	337	gate_30	38	True	False
2	377	gate_40	165	True	False
3	483	gate_40	1	False	False
4	488	gate_40	179	True	True

In [3]:

```
#数据概览
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 90189 entries, 0 to 90188
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   userid          90189 non-null  int64 
 1   version         90189 non-null  object
 2   sum_gamerounds  90189 non-null  int64 
 3   retention_1     90189 non-null  bool  
 4   retention_7     90189 non-null  bool  
dtypes: bool(2), int64(2), object(1)
memory usage: 2.2+ MB
```

In [4]:

```
#检查是否有空值
data.isnull().sum()
```

Out[4]:

```
userid            0
version           0
sum_gamerounds    0
retention_1       0
retention_7       0
dtype: int64
```

In [5]:

```
#检查是否有重复值
data.duplicated().sum()
```

Out[5]:

```
0
```

In [6]:

```
#检查是否有异常值
print('版　本　(值)：', data.version.unique())
print('次日留存率(值)：', data.retention_1.unique())
print('七日留存率(值)：', data.retention_7.unique())
```

```
版　本　(值)： ['gate_30' 'gate_40']
次日留存率(值)： [False  True]
七日留存率(值)： [False  True]
```

In [12]:

```
#数据清洗完成保存备份
data.to_csv('data_cleaned.csv')
```

4.数据分析

进行A/Btest**检验**的基本流程

1. 确定样本分布类型进行相关假设检验
2. 验证并得出结论

In [8]:

```
#提取分析所需数据
retention = data.groupby('version').agg({'userid':'count','retention_1':sum,'retention_7':sum})
retention
```

Out[8]:

version	userid	retention_1	retention_7
gate_30	44700	20034.0	8502.0
gate_40	45489	20119.0	8279.0

In [9]:

```
# 各方法的留存率
retention_rate = pd.DataFrame(index = ['gate_30','gate_40'],columns = ['次日留存率','七日留存率'])
for i in range(0,2):
    retention_rate.iloc[i,1] = retention.iloc[i,2] / retention.iloc[i,0]
    retention_rate.iloc[i,0] = retention.iloc[i,1] / retention.iloc[i,0]
retention_rate
```

Out[9]:

	次日留存率	七日留存率
gate_30	0.448188	0.190201
gate_40	0.442283	0.182

- 可以看出A方案（gate_30)无论是次日留存率还是七日留存率都要优于B方案（gate_40)
  接下来进行显著性测试确保我们的测试不是因为偶然因素。

a.假设检验：零假设和备择假设

设策略一概率为P1，策略二概率为P2，可得假设：
零假设H0：P1≥P2
备择假设Ha：P1<P2

b.分布类型、检验类型和显著性水平 ¶

样本服从二点分布，独立双样本，样本大小n > 30，总体均值和标准差未知，所以采用Z检验。显著性水平α取0.05。

In [10]:

```
z_score, p_value = ssp.proportions_ztest(count = retention.retention_1, nobs = retention.userid, alt
print("Z值为：", z_score)
print("P值为：", p_value)
```

```
Z值为： 1.7840862247974725
P值为： 0.9627951723515404
```

In [11]:

```
z_score, p_value = ssp.proportions_ztest(count = retention.retention_7, nobs = retention.userid, alt
print("Z值为：", z_score)
print("P值为：", p_value)
```

```
Z值为： 3.164358912748191
P值为： 0.9992228750121929
```

次日留存率P值为96.3%，七日留存率为99.9%，均大于(1-α == 0.95)，无法拒绝原假设。 所以我们可以得出结论**A方案（gate_30)的效果好于B方案（gate_40)**