CS146_Assignment_4

April 3, 2020

0.0.1 Johannes M. Halkenhaeusser

0.0.2 Minerva Schools at KGI

0.0.3 CS146 - Prof. Sheffler

0.0.4 Spring 2020

Model 1:

For each group (treatment and control), all 6 studies have the same fixed, but unknown, probability of success, $\theta_t, \theta_c \in [0, 1]$. The data follow a binomial distribution in each study, conditioned on the probability of success — θ_t for treatment or θ_c for control. The priors over θ_t and θ_c are uniform. These assumptions lead to the following model.

Likelihood: $\prod_{i=1}^{6} \text{Binomial}(s_i \mid \theta, n_i)$, where s_i is the number of successful recoveries, f_i is the number of failures (did not recover), and $n_i = s_i + f_i$ the number of patients.

Prior: $\text{Beta}(\theta \mid 1, 1)$ for both θ_t and θ_c.

Posterior for treatment group: $\text{Beta}(\theta_t \mid 108, 35)$.

Posterior for control group: $\text{Beta}(\theta_c \mid 58, 65)$.

What to do

The test statistic will be the standard deviation of the success rate. We'd expect the standard deviation to be smaller since each study - unlike this model assumes - should have a slightly different *theta*.

To test for this we will have to generate a sample θ from the control group posterior. For each study, this θ is fed into a binomial distribution that generates a sample for the number of successes given the studies participants.

[0]:
```
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as sts
```

[1]:
```
## From the data set provided in PCW

#no. successes
successes = [9,11,4,21,12,0]
```

```
#n
participants = [15,18,10,39,29,10]
```

[0]:
```
#define the test function

def test_function(data):
    #calculate the standard deviation of the success rates
    return np.std([data[i]/participants[i] for i in range(len(data))])
```

[39]:
```
#calculate the real test statistic
real_ts = test_function(successes)

## set up the main loop
p_val_count = 0

sample_size_theta = 10000
test_stats = []
for i in range(sample_size_theta):

    #sample a theta value
    theta = sts.beta.rvs(58,65)

    #sample the success number from the binomial distribution given theta for␣
↪each study
    sample_dataset = [sts.binom.rvs(n, theta) for n in participants]

    #calculate the test statistic
    test_stats.append(test_function(sample_dataset))

    #compare to the real test statistic
    if test_function(sample_dataset) > real_ts:
        p_val_count += 1

#calculate the p-value
p_value = p_val_count/sample_size_theta

print(f"The real test statistic is {real_ts}")
print(f"The mean test statistic of the reproduced data is {np.
↪mean(test_stats)}")
print(f"The p-value test statistic is {p_value}")
```

```
The real test statistic is 0.20795900518575364
The mean test statistic of the reproduced data is 0.108937485486605
The p-value test statistic is 0.008
```

[40]:
```
plt.hist(test_stats)
plt.axvline(test_function(successes), color = 'black', label = 'Real Test␣
↪Statistic')
plt.legend()
```

[40]: <matplotlib.legend.Legend at 0x7f29f501c080>

[40]:

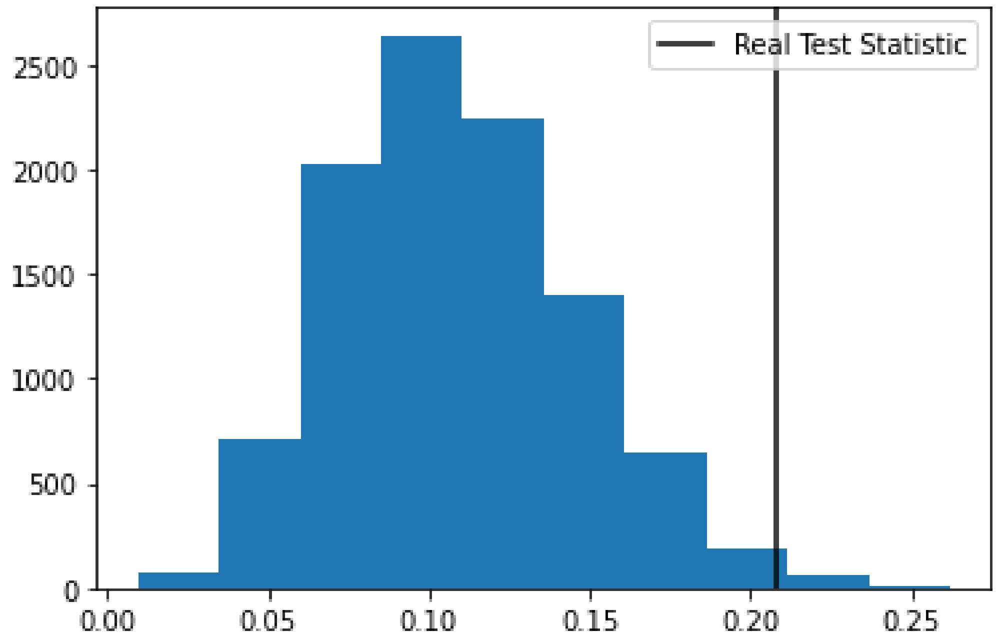

Figure 1. The real standard deviation of success rations compared to 10,000 generated samples.

Fig. 1 and the p-value show that in the first model, the assumption that the success rate is the same for each study is not mirrored in the control data as the spread of the success rates is statistically significantly higher in the real data.

[0]:
```

```