Examining Racial Discrimination in the US Job Market

Background

Racial discrimination continues to be pervasive in cultures throughout the world. Researchers examined the level of racial discrimination in the United States labor market by randomly assigning identical résumés to black-sounding or white-sounding names and observing the impact on requests for interviews from employers.

Data

In the dataset provided, each row represents a resume. The 'race' column has two values, 'b' and 'w', indicating black-sounding and white-sounding. The column 'call' has two values, 1 and 0, indicating whether the resume received a call from employers or not.

Note that the 'b' and 'w' values in race are assigned randomly to the resumes when presented to the employer.

Exercises

You will perform a statistical analysis to establish whether race has a significant impact on the rate of callbacks for resumes.

You can include written notes in notebook cells using Markdown:

- In the control panel at the top, choose Cell > Cell Type > Markdown
- Markdown syntax: http://nestacms.com/docs/creating-content/markdown-cheat-sheet (http://nestacms.com/docs/creating-content/markdown-cheat-sheet)

Resources

- Experiment information and data source: http://www.povertyactionlab.org/evaluation/discrimination-job-market-united-states (http://www.povertyactionlab.org/evaluation/discrimination-job-market-united-states)
- Scipy statistical methods: http://docs.scipy.org/doc/scipy/reference/stats.html (http://docs.scipy.org/doc/scipy/reference/stats.html)
- Markdown syntax: http://nestacms.com/docs/creating-content/markdown-cheat-sheet (http://nestacms.com/docs/creating-content/markdown-cheat-sheet)

Setup

This setups the data to study our hypotheses. We'll use numpy and scipy for our analysis.

In [1]:
```
import pandas as pd
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:
```
data = pd.io.stata.read_stata('data/us_job_market_discrimination.dta')
```

In [3]:
```
data.head()
```

Out[3]:

	id	ad	education	ofjobs	yearsexp	honors	volunteer	military	empholes	occupspecific	...	c
0	b	1	4	2	6	0	0	0	1	17	...	
1	b	1	3	3	6	0	1	1	0	316	...	
2	b	1	4	1	6	0	0	0	0	19	...	
3	b	1	3	4	6	0	1	0	1	313	...	
4	b	1	3	3	22	0	0	0	0	313	...	

5 rows × 65 columns

In [22]:
```
data.shape
```

Out[22]:
```
(4870, 65)
```

In [5]:
```
data.groupby('race').race.count()
```

Out[5]:
```
race
b    2435
w    2435
Name: race, dtype: int64
```

In [6]:
```
# number of callbacks by race
data.groupby('race').call.sum()
```

Out[6]:
```
race
b    157.0
w    235.0
Name: call, dtype: float32
```

In [7]:
```
da = data[data.race=='b'] # african americans
de = data[data.race=='w'] # european americans
```

Out of over 4,000 candidates, evenly divided between 'white'- and 'black'-sounding names, those with black-sounding names got 157 calls back, and those with white-sounding names got 235 calls. This implies that white candidates get more calls out of a sample of over 4,000 candidates. Let's investigate!

Strategy

What test would work here well? Would CLT work better here?

Multiple tests would work well here. Because this is a sample and with unknown population parameters, let's use the t-test and if time calls, we'll look into other tests. For Central Limit Theorem to apply:

1. The data must be independent from another: Yes, the data is independent from one another. With the null hypothesis, keep in mind that one set of candidates do not affect physically or emotionally another, only recruiting does.
2. At least 10 successes and 10 failures. In this case, at least 10 African Americans got callbacks, and so did their counterparts so this applies.

The Central Limit Theorem does apply here. The question is what type of test will work well here!

In [8]:
```
amu, asigma, an = np.mean(da.call), np.std(da.call), len(da)
amu, asigma, an
```

Out[8]: (0.0644763857126236, 0.24559901654720306, 2435)

In [9]:
```
emu, esigma, en = np.mean(de.call), np.std(de.call), len(de)
emu, esigma, en
```

Out[9]: (0.09650924056768417, 0.29528486728668213, 2435)

In [10]:
```
call_perc = sum(data.call)/len(data)
call_perc
```

Out[10]: 0.080492813141683772

Out of total callbacks, between blacks and whites is 1:2. On the contrary, the total percent of candidates who get callbacks is 8%. These are a good starting point to investigate. Let's set up the hypothesis.

Hypothesis Testing

What is the hypothesis and how are we going to test it?

$H_0 : \overline{X}_a = \overline{X}_e$

OR $\overline{X}_e - \overline{X}_a = 0$

$H_a : \overline{X}_a != \overline{X}_e$

Because we are dealing with categorical data and binary data (call or no, 1 or 0) a **two-sample T test** is a good runner, especially considering unknown population parameters! However, let's also try another test to validate this, maybe like the infering categorical variables as described in the *Open Statistics* 3rd Edition book.

Standard error: $SE_{\hat{p}} = \sqrt{\frac{p_1(1-p_1)}{n_1} + \frac{p_2(1-p_2)}{n_2}}$

In [11]:
```
# conditionals and difference of means (see Open Statistics 3rd Edition,
diff_mu = abs(emu - amu)
emu, amu, diff_mu
```

Out[11]: (0.09650924056768417, 0.0644763857126236, 0.03203285485506058)

In [12]:
```
# standard error for confidence interval
std_err_diff = np.sqrt(amu*(1-amu)/an + emu*(1-emu)/en)
std_err_diff
```

Out[12]: 0.0077833705860634299

In [13]:
```
# degrees of freedom
v = np.square((np.square(esigma)/en)+(np.square(asigma)/an))/(np.square(
v
```

Out[13]: 4711.6173117661901

In [14]:
```
# z-score for 95% percentile, using T-table calculator (http://stattrek.
zs = 1.96
```

In [15]:
```
# check with statistic and p-value
cat_prob_stat = diff_mu / std_err_diff
p_values_1=stats.norm.sf(abs(cat_prob_stat)) * 2
cat_prob_stat, p_values_1
```

Out[15]: (4.1155505190022987, 3.8625638129096897e-05)

In [16]:
```
# Confidence interval for 95%
ME3 = zs * std_err_diff
CI3 = [diff_mu - ME3, diff_mu + ME3]
ME3, CI3
```

Out[16]: (0.015255406348684322, [0.016777448506376254, 0.047288261203744901])

What about the two-sample T-Test? What does this say?

In [17]:
```
# manual 2-sample t-test
s_q = np.sqrt((np.square(esigma)/en)+(np.square(asigma)/an))
t_stat_2 = diff_mu / s_q
s_q, t_stat_2
```

Out[17]: (0.0077833083599234149, 4.1155834220829677)

In [18]:
```
p_values_2 = stats.norm.sf(abs(t_stat_2)) * 2 # one-sided
p_values_2
```

Out[18]: 3.8620128019721722e-05

In [19]:
```
# margin of error and confidence interval
ME2 = s_q * zs
CI2 = [((emu-amu) - ME2), ((emu-amu) + ME2)]
ME2, CI2
```

Out[19]: (0.015255284385449893, [0.016777570469610682, 0.047288139240510473])

In [20]:
```
# t-test via scipy
stats.ttest_ind(de.call, da.call)
```

Out[20]: Ttest_indResult(statistic=4.1147052908617514, pvalue=3.9408021031288859e-05)

To summarize this, we are 95% confident that the difference between candidates' callback will occur 0.01 to 0.48. In other words, rarely will there be an equilibrium between either race. And with scipy we see that the **p-value easily rejects the null hypothesis**. There is a difference and a concern that the average number of candidates by race were not equal.

Statistical Significance

The big picture concern is whether there is racial discrimination in the workplace. One hypothesis would be the callbacks, or return calls a candidate receives when they submit. However, this is one piece of the puzzle. Performing a categorical analysis implies a p-value of 3.85e-5, which is statistically significant. This means that we may reject the null hypothesis. If there is little to no discrimination, then the mean callbacks between races would be equal. However, it is not. The difference, in favor of European Americans, shows 95% confidence that they would get between 1% to 50%. This validates the earlier (and quicker claim) that European Americans are twice as likely to get callbacks than African Americans. However, callback is one story to the puzzle!

Conclusion

Does your analysis mean that race/name is the most important factor in callback success? Why or why not? If not, how would you amend your analysis?

Of course not! Employment by discrimination is a macroeconomic issue but it goes way beyond a person's name. For starters, while callbacks are a positive indication that the employer is interested in the person, it doesn't mean that the person got hired. Callbacks could range from 'screeing' to 'interviewing' to 'offers.' These vary.

Second, the dataset's attributes provide a deeper picture as they may have more information to digest and reflect. With a size of 65 columns, this implies that there was more to the employer's decision making process. These may include skillset, behaviorals (i.e. whether the manager prefers this candidate because they are easy to work with), and everything in between. They could imply different types of jobs that the company was planning to hire and how skills, background, and credentials may have played a role.

If I would have done the analysis again, I would wonder whether the military, education, and years of experience would have played a role in the hiring between whites and blacks. Some may raise eyebrowns. For instance, it's assumed those who have a military background would be looked down upon more than those without, as they could fester mental illness issues, such as PTSD. Other values would divide it based on job types; whether it is equal count that the number of black managers are equivalent to the number of white ones. Of course, this would dive deeper into the specifics, but that would be worth investigating!

In [ ]:

```
1
```