IDS: 690: Unifying Data Science

05FEB20 - In Class Exercise: Resume Experiment Analysis

Derek Wales, MIDS 21 and Joe Littell, MIDS 20

In [1]:
```
import pandas as pd
import numpy as np
import statsmodels.api as sm
import statsmodels.formula.api as smf
from statsmodels.formula.api import ols
import patsy
from plotnine import *
from scipy.stats import ttest_ind
```

In [2]:
```
resume_df = pd.read_stata('resume_experiment.dta')
```

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

Out[3]:

	education	ofjobs	yearsexp	computerskills	call	female	black
0	4	2	6	1	0.0	1.0	0.0
1	3	3	6	1	0.0	1.0	0.0
2	4	1	6	1	0.0	1.0	1.0
3	3	4	6	1	0.0	1.0	1.0
4	3	3	22	1	0.0	1.0	0.0

Exercise 1: Balance Check

In [4]:
```
white_resume_df = resume_df.loc[(resume_df['black'] == 0)]
white_resume_df.head(1)
```

Out[4]:

	education	ofjobs	yearsexp	computerskills	call	female	black
0	4	2	6	1	0.0	1.0	0.0

In [5]:
```
black_resume_df = resume_df.loc[(resume_df['black'] == 1)]
black_resume_df.head(1)
```

Out[5]:

	education	ofjobs	yearsexp	computerskills	call	female	black
2	4	1	6	1	0.0	1.0	1.0

In [6]:
```
resume_df[resume_df['black']==1].describe()
```

Out[6]:

	education	ofjobs	yearsexp	computerskills	call	female	black
count	2435.000000	2435.000000	2435.000000	2435.000000	2435.000000	2435.000000	2435.0
mean	3.616016	3.658316	7.829569	0.832444	0.064476	0.774538	1.0
std	0.733060	1.219150	5.010764	0.373549	0.245649	0.417974	0.0
min	0.000000	1.000000	1.000000	0.000000	0.000000	0.000000	1.0
25%	3.000000	3.000000	5.000000	1.000000	0.000000	1.000000	1.0
50%	4.000000	4.000000	6.000000	1.000000	0.000000	1.000000	1.0
75%	4.000000	4.000000	9.000000	1.000000	0.000000	1.000000	1.0
max	4.000000	7.000000	44.000000	1.000000	1.000000	1.000000	1.0

In [7]:
```
resume_df[resume_df['black']==0].describe()
```

Out[7]:

	education	ofjobs	yearsexp	computerskills	call	female	black
count	2435.000000	2435.000000	2435.000000	2435.000000	2435.000000	2435.000000	2435.0
mean	3.620945	3.664476	7.856263	0.808624	0.096509	0.763860	0.0
std	0.696609	1.219345	5.079228	0.393465	0.295346	0.424794	0.0
min	0.000000	1.000000	1.000000	0.000000	0.000000	0.000000	0.0
25%	3.000000	3.000000	5.000000	1.000000	0.000000	1.000000	0.0
50%	4.000000	4.000000	6.000000	1.000000	0.000000	1.000000	0.0
75%	4.000000	4.000000	9.000000	1.000000	0.000000	1.000000	0.0
max	4.000000	7.000000	26.000000	1.000000	1.000000	1.000000	0.0

Both datasets look similar, we will validate with the T_Test.

In [8]:
```
# Computer Skills for Whites
ttest_ind(white_resume_df[white_resume_df['computerskills'] == 0].call.values,
white_resume_df[white_resume_df['computerskills'] == 1].call.values)
```

Out[8]:
```
Ttest_indResult(statistic=1.9243517330441888, pvalue=0.054427013910828013)
```

In [9]:
```
# Computer Skills for Blacks
ttest_ind(black_resume_df[black_resume_df['computerskills'] == 0].call.values,
black_resume_df[black_resume_df['computerskills'] == 1].call.values)
```

Out[9]:
```
Ttest_indResult(statistic=0.5949176664034509, pvalue=0.5519538306383964)
```

In [10]:
```
# Female Whites
ttest_ind(white_resume_df[white_resume_df['female'] == 0].call.values, white_r
esume_df[white_resume_df['female'] == 1].call.values)
```

Out[10]:
```
Ttest_indResult(statistic=-0.7257712889249252, pvalue=0.4680487958977867)
```

In [11]:
```
# Female Blacks
ttest_ind(black_resume_df[black_resume_df['female'] == 0].call.values, black_r
esume_df[black_resume_df['female'] == 1].call.values)
```

Out[11]:
```
Ttest_indResult(statistic=-0.6706419018702243, pvalue=0.5025123365847134)
```

In [12]:
```
# Education whites
ttest_ind(white_resume_df[white_resume_df['education'] > 3.5].call.values, whi
te_resume_df[white_resume_df['education'] <= 3.5].call.values)
```

Out[12]:
```
Ttest_indResult(statistic=-0.8084223548693196, pvalue=0.4189265258048491)
```

In [13]:
```
# Education blacks
ttest_ind(black_resume_df[black_resume_df['education'] > 3.5].call.values, bla
ck_resume_df[black_resume_df['yearsexp'] <= 3.5].call.values)
```

Out[13]:
```
Ttest_indResult(statistic=0.1877576824876979, pvalue=0.8510852010748955)
```

In [14]:
```
# Years Experience
ttest_ind(white_resume_df[white_resume_df['yearsexp'] > 7.5].call.values, whit
e_resume_df[white_resume_df['yearsexp'] <= 7.5].call.values)
```

Out[14]:
```
Ttest_indResult(statistic=2.5937008790361364, pvalue=0.009551844650430422)
```

In [15]:
```
# Years Experience is a significant factor
ttest_ind(black_resume_df[black_resume_df['yearsexp'] > 7.5].call.values, blac
k_resume_df[black_resume_df['yearsexp'] <= 7.5].call.values)
```

Out[15]:
```
Ttest_indResult(statistic=2.974966145673845, pvalue=0.0029590457543845552)
```

Exercise 2: Education Distribution

Source: https://pythonfordatascience.org/chi-square-test-of-independence-python/ (https://pythonfordatascience.org/chi-square-test-of-independence-python/)

In [16]:
```
from scipy import stats
from scipy.stats import chisquare
```

In [17]:
```
crosstab = pd.crosstab(resume_df['black'], resume_df['education'])
crosstab
```

Out[17]:

education	0	1	2	3	4
black					
0.0	18	18	142	513	1744
1.0	28	22	132	493	1760

Running the chi squared test across education and black.

In [18]:
```
stats.chi2_contingency(crosstab)
```

Out[18]:
```
(3.4095502219737974,
 0.4917640058792273,
 4,
 array([[  23.,    20.,   137.,   503., 1752.],
        [  23.,    20.,   137.,   503., 1752.]]))
```

P value is .49, which is above .05. Meaning we accept the null hypothesis that the distributions are the same (aka no significant differences between blacks and whites in terms of education).

Exercise 3: Results of resume characteristics

The overall characteristics are balanced across terms which is what we want because it shows that there is no noticible baseline difference

Exercise 4: Determining if black appicants were called back

In [19]:
```
# T-test of call
ttest_ind(resume_df[resume_df['black'] == 0].call.values, resume_df[resume_df[
'black'] == 1].call.values)
```

Out[19]:
```
Ttest_indResult(statistic=4.114705290861751, pvalue=3.940802103128886e-05)
```

Having a black sounding name is hugely influential in determining if you are called back.

Exercise 5: Linear Regression

In [20]:
```
smf.ols('call ~ black', data = resume_df).fit().summary()
```

Out[20]:

OLS Regression Results

Dep. Variable:	call	**R-squared:**	0.003
Model:	OLS	**Adj. R-squared:**	0.003
Method:	Least Squares	**F-statistic:**	16.93
Date:	Wed, 12 Feb 2020	**Prob (F-statistic):**	3.94e-05
Time:	00:02:33	**Log-Likelihood:**	-562.24
No. Observations:	4870	**AIC:**	1128.
Df Residuals:	4868	**BIC:**	1141.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	0.0965	0.006	17.532	0.000	0.086	0.107
black	-0.0320	0.008	-4.115	0.000	-0.047	-0.017

Omnibus:	2969.205	**Durbin-Watson:**	1.440
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	18927.068
Skew:	3.068	**Prob(JB):**	0.00
Kurtosis:	10.458	**Cond. No.**	2.62

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Exercise 6: With control variables

In [21]:
```
smf.ols('call ~ black + yearsexp + female + computerskills + C(education) + of
jobs', data = resume_df).fit().summary()
```

Out[21]:

OLS Regression Results

Dep. Variable:	call	**R-squared:**	0.008
Model:	OLS	**Adj. R-squared:**	0.006
Method:	Least Squares	**F-statistic:**	4.445
Date:	Wed, 12 Feb 2020	**Prob (F-statistic):**	8.01e-06
Time:	00:02:33	**Log-Likelihood:**	-550.73
No. Observations:	4870	**AIC:**	1121.
Df Residuals:	4860	**BIC:**	1186.
Df Model:	9		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	0.0860	0.042	2.036	0.042	0.003	0.169
C(education)[T.1]	0.0006	0.059	0.011	0.991	-0.115	0.116
C(education)[T.2]	0.0022	0.044	0.051	0.959	-0.083	0.088
C(education)[T.3]	0.0009	0.041	0.021	0.983	-0.080	0.082
C(education)[T.4]	-0.0005	0.041	-0.013	0.990	-0.080	0.079
black	-0.0316	0.008	-4.066	0.000	-0.047	-0.016
yearsexp	0.0033	0.001	4.101	0.000	0.002	0.005
female	0.0104	0.010	1.058	0.290	-0.009	0.030
computerskills	-0.0175	0.011	-1.628	0.104	-0.039	0.004
ofjobs	-0.0026	0.003	-0.753	0.451	-0.009	0.004

Omnibus:	2949.995	**Durbin-Watson:**	1.448
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	18619.766
Skew:	3.046	**Prob(JB):**	0.00
Kurtosis:	10.392	**Cond. No.**	240.

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Adding the additional controls improved both the Adjusted R^2 and regular R^2

Exercise 7

In [22]:
```
highed_resume_df = resume_df[resume_df['education'] == 4]
```

In [23]:
```
smf.ols('call ~ black + yearsexp + female + computerskills + C(education) + of
jobs', data = highed_resume_df).fit().summary()
```

Out[23]:

OLS Regression Results

Dep. Variable:	call	**R-squared:**	0.006
Model:	OLS	**Adj. R-squared:**	0.004
Method:	Least Squares	**F-statistic:**	3.952
Date:	Wed, 12 Feb 2020	**Prob (F-statistic):**	0.00142
Time:	00:02:34	**Log-Likelihood:**	-371.86
No. Observations:	3504	**AIC:**	755.7
Df Residuals:	3498	**BIC:**	792.7
Df Model:	5		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	0.0885	0.019	4.581	0.000	0.051	0.126
black	-0.0287	0.009	-3.155	0.002	-0.047	-0.011
yearsexp	0.0022	0.001	2.184	0.029	0.000	0.004
female	0.0179	0.010	1.718	0.086	-0.003	0.038
computerskills	-0.0082	0.012	-0.680	0.496	-0.032	0.015
ofjobs	-0.0048	0.004	-1.247	0.212	-0.012	0.003

Omnibus:	2164.428	**Durbin-Watson:**	1.522
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	14132.777
Skew:	3.094	**Prob(JB):**	0.00
Kurtosis:	10.649	**Cond. No.**	44.4

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

For well educated canidates the descrimination against blacks decreases slightly from reducing your odds of getting called back from 3.16 to 2.87%.

Exercise 8: Comparison between black men and black women

In [24]:
```
smf.ols('call ~ black * female + yearsexp + computerskills + C(education) + of
jobs', data = highed_resume_df).fit().summary()
```

Out[24]:

OLS Regression Results

Dep. Variable:	call	**R-squared:**	0.006
Model:	OLS	**Adj. R-squared:**	0.004
Method:	Least Squares	**F-statistic:**	3.293
Date:	Wed, 12 Feb 2020	**Prob (F-statistic):**	0.00311
Time:	00:02:34	**Log-Likelihood:**	-371.86
No. Observations:	3504	**AIC:**	757.7
Df Residuals:	3497	**BIC:**	800.8
Df Model:	6		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	0.0880	0.021	4.269	0.000	0.048	0.128
black	-0.0276	0.017	-1.611	0.107	-0.061	0.006
female	0.0186	0.014	1.294	0.196	-0.010	0.047
black:female	-0.0015	0.020	-0.076	0.939	-0.041	0.038
yearsexp	0.0022	0.001	2.184	0.029	0.000	0.004
computerskills	-0.0082	0.012	-0.680	0.496	-0.032	0.015
ofjobs	-0.0048	0.004	-1.247	0.212	-0.012	0.003

Omnibus:	2164.432	**Durbin-Watson:**	1.522
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	14132.988
Skew:	3.094	**Prob(JB):**	0.00
Kurtosis:	10.649	**Cond. No.**	64.6

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Black men are discriminated against more than black women. Being a black man reduces your odds of being called back by 2.76% but being a black woman it is only 0.15%.

Exercise 9: Our study vs. population data

In [25]:
```
# Americans with college degrees
resume_df['education'].value_counts(normalize = True)
```

Out[25]:
```
4    0.719507
3    0.206571
2    0.056263
0    0.009446
1    0.008214
Name: education, dtype: float64
```

In [26]:
```
# Black Americans with college degrees
resume_df[resume_df['black'] == 1]['education'].value_counts(normalize = True)
```

Out[26]:
```
4    0.722793
3    0.202464
2    0.054209
0    0.011499
1    0.009035
Name: education, dtype: float64
```

Exercise 10: What are your answers to the regressions

Being black makes it between 4.7 and 11 % less likely to be called backed when controlling for all other factors such as education, years experience, gender, and number of jobs.

For being black as a treatment effect, we see a greater effect, between 4.7 and 16% less likely to be called back.