Case Study: Chi-test and post-hoc test for association between HIV rate and employment rate

Summary:

First, I run chi-test for association between HIV rate and employment rate, recorded as table of 2 rows by 4 columns. The HIV rate group 1 means that people are not infected with HIV. The HIV rate group 2 means that people infected with HIV. Employment rate is separated in 4 categorical groups: group 1: employment is between 31-44%; group 2: rate is between 45-57%; group 3: rate is between 58-70% and group4: rate is between 71-84%.
The chi-square value equals to 11.54 and p value equals to 0.009 (< 0.05). This suggests to reject a null hypothesis that all values are different.

Next, I run the post-hoc test on 6 pair comparisons (1-2, 1-3, 1-4, 2-3, 2-4) to identify which pairs are different one from another. The adjusted p value equals 0.05/6=0.00833.

Pairs:	p-value:	conclusion:
1-2	0.166 > 0.00833	not different, then except null hypothesis
1-3	0.910 > 0.00833	not different, then except null hypothesis
1-4	0.628 > 0.00833	not different, then except null hypothesis
2-3	0.110 > 0.00833	not different, then except null hypothesis
2-4	0.005 < 0.00833	different, then reject null hypothesis
3-4	0.311 > 0.00833	not different, then except null hypothesis

The post-hoc test shows that majority of groups are not different from each other, thus we should expect the null hypothesis, which is different from the conclusion of the original chi-test.

Output:

ecgroup4	1	2	3	4
hcgroup4				
1	1	19	8	2
2	2	3	6	6

ecgroup4	1	2	3	4
hcgroup4				
1	0.333333	0.863636	0.571429	0.250000

2 0.666667 0.136364 0.428571 0.750000

chi-square value, p value, expected counts
(11.542746795687972, 0.0091255175001177923, 3, array([[1.91489362,
14.04255319, 8.93617021, 5.10638298],
[1.08510638, 7.95744681, 5.06382979, 2.89361702]]))

comp_var_12 1.000000 2.000000
hcgroup4
1 1 19
2 2 3

comp_var_12 1.000000 2.000000
hcgroup4
1 0.333333 0.863636
2 0.666667 0.136364

chi-square value for 1-2 pair, p value, expected counts
(1.9176136363636365, 0.16611997044958737, 1, array([[2.4, 17.6],
[0.6, 4.4]]))
comp_var_13 1.000000 3.000000
hcgroup4
1 1 8
2 2 6
comp_var_13 1.000000 3.000000
hcgroup4
1 0.333333 0.571429
2 0.666667 0.428571

chi-square value for 1-3 pair, p value, expected counts
(0.012648809523809488, 0.91045319227742638, 1, array([[1.58823529,
7.41176471], [1.41176471, 6.58823529]]))

comp_var_14 1.000000 4.000000
hcgroup4
1 1 2
2 2 6
comp_var_14 1.000000 4.000000
hcgroup4
1 0.333333 0.250000

2 0.666667 0.750000

chi-square value for 1-4 pair, p value, expected counts
(0.23394097222222238, 0.62861692967057459, 1, array([[0.81818182, 2.18181818], [2.18181818, 5.81818182]]))

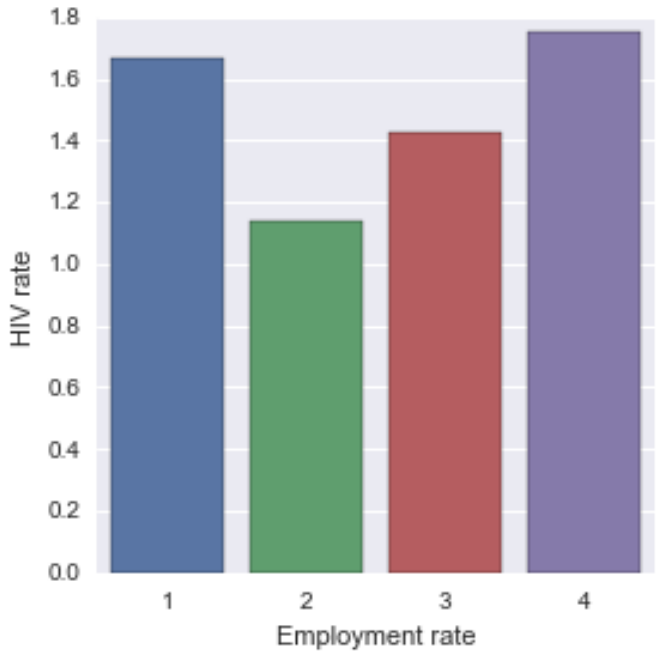
comp_var_23 2.000000 3.000000
hcgroup4
1 19 8
2 3 6
comp_var_23 2.000000 3.000000
hcgroup4
1 0.863636 0.571429
2 0.136364 0.428571

chi-square value for 2-3 pair, p value, expected counts
(2.4935064935064934, 0.11431677554040424, 1, array([[16.5, 10.5],
[5.5, 3.5]]))
comp_var_24 2.000000 4.000000
hcgroup4
1 19 2
2 3 6
comp_var_24 2.000000 4.000000
hcgroup4
1 0.863636 0.250000
2 0.136364 0.750000

chi-square value for 2-4 pair, p value, expected counts
(7.8003246753246751, 0.0052236847465871317, 1, array([[15.4, 5.6], [6.6, 2.4]]))
comp_var_34 3.000000 4.000000
hcgroup4
1 8 2
2 6 6
comp_var_34 3.000000 4.000000
hcgroup4
1 0.571429 0.250000
2 0.428571 0.750000

chi-square value for 3-4 pair, p value, expected counts

(1.0230654761904767, 0.31179297156691088, 1, array([[6.36363636,
3.63636364], [7.63636364, 4.36363636]]))



Python Program

```
"""
Created on Tue Oct 27 2015

@author: violetgirl
"""
import pandas as pd
import numpy as np
import scipy.stats
import seaborn as sb
import matplotlib.pyplot as plt

# load gapminder dataset
data = pd.read_csv('gapminder.csv',low_memory=False)
# lower-case all DataFrame column names
data.columns = map(str.lower, data.columns)
# bug fix for display formats to avoid run time errors
pd.set_option('display.float_format', lambda x:'%f'%x)
```

```
# setting variables to be numeric
data['suicideper100th'] =
data['suicideper100th'].convert_objects(convert_numeric=True)
data['breastcancerper100th'] =
data['breastcancerper100th'].convert_objects(convert_numeric=True)
data['hivrate'] = data['hivrate'].convert_objects(convert_numeric=True)
data['employrate'] = data['employrate'].convert_objects(convert_numeric=True)

# display summary statistics about the data
#print("Statistics for a Suicide Rate")
#print(data['suicideper100th'].describe())

# subset data for a high suicide rate based on summary statistics
sub = data[(data['suicideper100th']>12)]
#make a copy of my new subsetted data
sub_copy = sub.copy()
# remove missing values
sub_copy=sub_copy.dropna()

# EMPLOYMENT RATE
# group the data in 4 groups and record it into new variable ecgroup4
def ecgroup4 (row):
    if row['employrate'] >= 32 and row['employrate'] < 45:
        return 1
    elif row['employrate'] >= 45 and row['employrate'] < 58:
        return 2
    elif row['employrate'] >= 58 and row['employrate'] < 71:
        return 3
    elif row['employrate'] >= 71 and row['employrate'] < 84:
        return 4

sub_copy['ecgroup4'] = sub_copy.apply(lambda row:  ecgroup4 (row), axis=1)

# HIV RATE
# group the data in 2 groups and record it into new variable hcgroup4

def hcgroup4 (row):
    if row['hivrate'] >= 0 and row['hivrate'] < 1 :
        return 1 # not infected with HIV
    elif row['hivrate'] >= 1 and row['hivrate'] < 26:
```

```
        return 2 # infected with HIV
sub_copy['hcgroup4'] = sub_copy.apply(lambda row:  hcgroup4 (row), axis=1)

# contingency table of observed counts
ct1=pd.crosstab(sub_copy['hcgroup4'], sub_copy['ecgroup4'])
print (ct1)

# column percentages
colsum=ct1.sum(axis=0)
colpct=ct1/colsum
print(colpct)

# chi-square
print ('chi-square value, p value, expected counts')
cs1= scipy.stats.chi2_contingency(ct1)
print (cs1)

# plot results to understand a relationship
# set explanatory variable to categoric
sub_copy['ecgroup4']=sub_copy['ecgroup4'].astype('category')
# set response variable to numeric
sub_copy['hcgroup4']=sub_copy['hcgroup4'].convert_objects(convert_numeric=T
rue)

sb.factorplot(x='ecgroup4',y='hcgroup4',data=sub_copy,kind="bar",ci=None)
plt.xlabel("Employment rate")
plt.ylabel("HIV rate")

# Post-hoc test
# Chi test for the first pair 1-2
recode2 = {1: 1, 2: 2}
sub_copy['comp_var_12']= sub_copy['ecgroup4'].map(recode2)
# contingency table of observed counts
ct1=pd.crosstab(sub_copy['hcgroup4'], sub_copy['comp_var_12'])
print (ct1)
# column percentages
colsum=ct1.sum(axis=0)
colpct=ct1/colsum
print(colpct)
# chi-square
```

```
print ('chi-square value for 1-2 pair, p value, expected counts')
cs1= scipy.stats.chi2_contingency(ct1)
print (cs1)

# Chi test for the first pair 1-3
recode2 = {1: 1, 3: 3}
sub_copy['comp_var_13']= sub_copy['ecgroup4'].map(recode2)
# contingency table of observed counts
ct1=pd.crosstab(sub_copy['hcgroup4'], sub_copy['comp_var_13'])
print (ct1)
# column percentages
colsum=ct1.sum(axis=0)
colpct=ct1/colsum
print(colpct)
# chi-square
print ('chi-square value for 1-3 pair, p value, expected counts')
cs1= scipy.stats.chi2_contingency(ct1)
print (cs1)

# Chi test for the first pair 1-4
recode2 = {1: 1, 4: 4}
sub_copy['comp_var_14']= sub_copy['ecgroup4'].map(recode2)
# contingency table of observed counts
ct1=pd.crosstab(sub_copy['hcgroup4'], sub_copy['comp_var_14'])
print (ct1)
# column percentages
colsum=ct1.sum(axis=0)
colpct=ct1/colsum
print(colpct)
# chi-square
print ('chi-square value for 1-4 pair, p value, expected counts')
cs1= scipy.stats.chi2_contingency(ct1)
print (cs1)

# Chi test for the first pair 2-3
recode2 = {2: 2, 3: 3}
sub_copy['comp_var_23']= sub_copy['ecgroup4'].map(recode2)
# contingency table of observed counts
ct1=pd.crosstab(sub_copy['hcgroup4'], sub_copy['comp_var_23'])
print (ct1)
```

```
# column percentages
colsum=ct1.sum(axis=0)
colpct=ct1/colsum
print(colpct)
# chi-square
print ('chi-square value for 2-3 pair, p value, expected counts')
cs1= scipy.stats.chi2_contingency(ct1)
print (cs1)

# Chi test for the first pair 2-4
recode2 = {2: 2, 4: 4}
sub_copy['comp_var_24']= sub_copy['ecgroup4'].map(recode2)
# contingency table of observed counts
ct1=pd.crosstab(sub_copy['hcgroup4'], sub_copy['comp_var_24'])
print (ct1)
# column percentages
colsum=ct1.sum(axis=0)
colpct=ct1/colsum
print(colpct)
# chi-square
print ('chi-square value for 2-4 pair, p value, expected counts')
cs1= scipy.stats.chi2_contingency(ct1)
print (cs1)

# Chi test for the first pair 3-4
recode2 = {3: 3, 4: 4}
sub_copy['comp_var_34']= sub_copy['ecgroup4'].map(recode2)
# contingency table of observed counts
ct1=pd.crosstab(sub_copy['hcgroup4'], sub_copy['comp_var_34'])
print (ct1)
# column percentages
colsum=ct1.sum(axis=0)
colpct=ct1/colsum
print(colpct)
# chi-square
print ('chi-square value for 3-4 pair, p value, expected counts')
cs1= scipy.stats.chi2_contingency(ct1)
print (cs1)

# END
```