Assignment 4: Logistic regression for HIV rate vs. urbanization rate.

Hypothesis:
There is a strong correlation between the HIV rate and urban rate for 203 countries considered in my sample. I considered additional explanatory variables (one at a time), such as life expectancy and alcohol consumption, to identify potential confounders for this correlation.

Data preparation:

HIV rate is a quantitative response variable, therefore I bin into two categories: people without HIV (prevalence < 2%) and people with HIV (prevalence > 2%). Urbanization rate is a quantitative explanatory variable, I centered to its mean.
Potential confounding variables (life expectancy and alcohol consumption) are quantitative; I centered them to their mean. During the data preparation NAN values were removed from the sample, resulting in the size of the sample of 142 countries.

Results:

1) Case 1: no confounder included:

   results for urban rate: OR=0.96, 95% CI =0.94-0.98, p=0.000.

The p value < 0.0001 show that the urban rate is significantly associated with the HIV rate. However, the OR < 1 show that an increase of the urban rate is significantly less likely associated with the HIV rate in the current dataset.

2) Case 2: life expectancy confounder included:

   results for urban rate: OR=1.03, 95% CI =0.99-1.07, p=0.110;

   results for life expectancy: OR=0.76, 95% CI =0.69- 0.84, p=0.000.

The p value >>> 0.0001 for urban rate show that the urban rate is not associated with the HIV rate after including the confounder. The p value < 0.0001 for the life expectancy show that the life expectancy is significantly associated with the HIV rate. However, the OR < 1 show that an increase of the life expectancy is significantly less likely associated with the HIV rate in the current dataset after controlling for the urban rate.

3) Case 3: alcohol consumption confounder included:

   results for urban rate: OR= 0.96, 95% CI = 0.94-0.98, p= 0.000;

   results for alcohol consumption: OR= 1.03, 95% CI = 0.94 - 1.13, p= 0.526.

The p value >>> 0.0001 for alcohol consumption show that alcohol consumption is not associated with the HIV rate after including the confounder. The p value < 0.0001 for the urban rate show that the urban rate is significantly associated with the HIV rate. However, the OR < 1 show that an increase of the urban rate is significantly less likely associated with the HIV rate in the current dataset after controlling for alcohol consumption.

Summary: Current results show that there is no association between the HIV rate and urban rate in my sample (opposite of my original hypothesis), even after testing two additional confounding variables, such as life expectancy and alcohol consumption.

Python Output:

Logit Regression Results

Dep. Variable:	hiv_c	No. Observations:	142
Model:	Logit	Df Residuals:	140
Method:	MLE	Df Model:	1
Date:	Tue, 12 Jan 2016	Pseudo R-squ.:	0.1069
Time:	14:54:41	Log-Likelihood:	-62.962
converged:	True	LL-Null:	-70.499
		LLR p-value:	0.0001033

	coef	std err	z	P>|z|	[95.0% Conf.	Int.]
Intercept	-1.6237	0.252	-6.456	0.000	-2.117	-1.131
urbanrate	-0.0392	0.011	-3.597	0.000	-0.061	-0.018

Odds Ratios

Intercept	0.20
urbanrate	0.96

dtype: float64

	Lower CI	Upper CI	OR
Intercept	0.12	0.32	0.20
urbanrate	0.94	0.98	0.96

Logit Regression Results

Dep. Variable:	hiv_c	No. Observations:	142
Model:	Logit	Df Residuals:	139
Method:	MLE	Df Model:	2
Date:	Tue, 12 Jan 2016	Pseudo R-squ.:	0.5311
Time:	14:54:41	Log-Likelihood:	-33.060
converged:	True	LL-Null:	-70.499
		LLR p-value:	5.502e-17

	coef	std err	z	P>|z|	[95.0% Conf.	Int.]
Intercept	-2.6748	0.482	-5.553	0.000	-3.619	-1.731
urbanrate	0.0312	0.020	1.597	0.110	-0.007	0.070
lifeexpectancy	-0.2732	0.050	-5.488	0.000	-0.371	-0.176

	Lower CI	Upper CI	OR
Intercept	0.03	0.18	0.07
urbanrate	0.99	1.07	1.03
lifeexpectancy	0.69	0.84	0.76

```
                           Logit Regression Results
==============================================================================
Dep. Variable:                  hiv_c   No. Observations:                  142
Model:                          Logit   Df Residuals:                      139
Method:                           MLE   Df Model:                            2
Date:                Tue, 12 Jan 2016   Pseudo R-squ.:                  0.1097
Time:                        14:54:41   Log-Likelihood:                -62.762
converged:                       True   LL-Null:                       -70.499
                                        LLR p-value:                 0.0004363
==============================================================================
                     coef    std err          z      P>|z|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
Intercept         -1.6308      0.252     -6.460      0.000        -2.126    -1.136
urbanrate         -0.0419      0.012     -3.533      0.000        -0.065    -0.019
alcconsumption     0.0303      0.048      0.635      0.526        -0.063     0.124
==============================================================================
                Lower CI  Upper CI   OR
Intercept           0.12      0.32 0.20
urbanrate           0.94      0.98 0.96
alcconsumption      0.94      1.13 1.03
Optimization terminated successfully.
         Current function value: 0.432510
```

Python Code:

```
import numpy as np
import pandas as pd
import statsmodels.formula.api as smf
import statsmodels.stats.multicomp as multi
import scipy
import scipy.stats
import statsmodels.api
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:'%.2f'%x)

# remove missing values
data_copy=data.copy()
data_copy=data_copy.dropna()

# convert variables to numeric format using convert_objects function
# quantitative reponse variable - HIV rate; quantitative explanatory variable - urbanrate
data_copy['hivrate'] = data_copy['hivrate'].convert_objects(convert_numeric=True)
data_copy['urbanrate'] = data_copy['urbanrate'].convert_objects(convert_numeric=True)
# additional quantitative explanatory variables
data_copy['lifeexpectancy'] = data_copy['lifeexpectancy'].convert_objects(convert_numeric=True)
data_copy['alcconsumption'] = data_copy['alcconsumption'].convert_objects(convert_numeric=True)

# Data preparation
```

```
sub1 = data_copy[['hivrate','urbanrate','lifeexpectancy','alcconsumption']]
sub1=sub1.dropna()

# bin the response variable - HIV rate in two categories
def hiv_c(row):
    if row['hivrate'] < sub1['hivrate'].mean():
        return 0
    else:
        return 1
sub1['hiv_c'] = sub1.apply(lambda row:  hiv_c (row), axis=1)

# center all quantitative explanatory variables
sub1['urbanrate']=sub1['urbanrate']-sub1['urbanrate'].mean()
sub1['lifeexpectancy']=sub1['lifeexpectancy']-sub1['lifeexpectancy'].mean()
sub1['alcconsumption']=sub1['alcconsumption']-sub1['alcconsumption'].mean()

# Logistic regression with urbanrate only: no confounder uncluded
lreg1 = smf.logit(formula = 'hiv_c ~ urbanrate', data = sub1).fit()
print (lreg1.summary())
# odds ratios
print ("Odds Ratios")
print (np.exp(lreg1.params))

# Odd ratios with 95% confidence intervals
params = lreg1.params
conf = lreg1.conf_int()
conf['OR'] = params
conf.columns = ['Lower CI', 'Upper CI', 'OR']
print (np.exp(conf))

# logistic regression with urbanrate and potential confounder: lifeexpectancy
lreg2 = smf.logit(formula = 'hiv_c ~ urbanrate + lifeexpectancy', data = sub1).fit()
print (lreg2.summary())

# odd ratios with 95% confidence intervals
params = lreg2.params
conf = lreg2.conf_int()
conf['OR'] = params
conf.columns = ['Lower CI', 'Upper CI', 'OR']
print (np.exp(conf))

# logistic regression with urbanrate and potential confounder: alcconsumption
lreg4 = smf.logit(formula = 'hiv_c ~ urbanrate + alcconsumption', data = sub1).fit()
print (lreg4.summary())

# odd ratios with 95% confidence intervals
params = lreg4.params
conf = lreg4.conf_int()
conf['OR'] = params
conf.columns = ['Lower CI', 'Upper CI', 'OR']
print (np.exp(conf))
```