Practical 9 - JosiahTeh

January 3, 2022

1 First Name: Josiah

2 Last Name: Teh

3 Import Libraries

[1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import statsmodels.formula.api as smf
```

4 Read in data

[2]:
```
nesarc = pd.read_csv('nesarc - large.csv', low_memory=False)
pd.set_option('display.float_format', lambda x:'%f'%x)
```

[3]:
```
nesarc['S2AQ5B'] = pd.to_numeric(nesarc['S2AQ5B'], errors='coerce') #convert␣
 ↪variable to numeric
nesarc['S2AQ5D'] = pd.to_numeric(nesarc['S2AQ5D'], errors='coerce') #convert␣
 ↪variable to numeric
nesarc['S2AQ5A'] = pd.to_numeric(nesarc['S2AQ5A'], errors='coerce') #convert␣
 ↪variable to numeric
nesarc['S2BQ1B1'] = pd.to_numeric(nesarc['S2BQ1B1'], errors='coerce') #convert␣
 ↪variable to numeric
nesarc['AGE'] = pd.to_numeric(nesarc['AGE'], errors='coerce') #convert variable␣
 ↪to numeric
```

[4]:
```
sub1=nesarc[(nesarc['AGE']>=26) & (nesarc['AGE']<=50) & (nesarc['S2AQ5A']==1)]
sub2=sub1.copy()
```

[5]:
```
sub2['S2AQ5D']=sub2['S2AQ5D'].replace(99, np.nan)

sub2['S2AQ5B']=sub2['S2AQ5B'].replace(8, np.nan)
sub2['S2AQ5B']=sub2['S2AQ5B'].replace(9, np.nan)
sub2['S2AQ5B']=sub2['S2AQ5B'].replace(10, np.nan)
```

```
sub2['S2AQ5B']=sub2['S2AQ5B'].replace(99, np.nan)

sub2['S2BQ1B1']=sub2['S2BQ1B1'].replace(9, np.nan)
```

[6]:
```
recode2 = {1:30, 2:26, 3:14, 4:8, 5:4, 6:2.5, 7:1}
sub2['BEER_FEQMO']= sub2['S2AQ5B'].map(recode2)

recode3 = {2:0, 1:1}
sub2['S2BQ1B1']= sub2['S2BQ1B1'].map(recode3)
```

[7]:
```
#secondary variable
sub2['NUMBEERMO_EST']=sub2['BEER_FEQMO'] * sub2['S2AQ5D']
```

5 Scenario 1

6 Perform Regression analysis between

7 Beer dependency (S2BQ1B1 - Categorical Explanatory variable) and

8 number of beers consumed in a month (NUMBEERMO_EST - Quantitative Response variable)

9 use sub2

[8]:
```
# hint cell 9
reg1 = smf.ols('NUMBEERMO_EST ~ S2BQ1B1', data=sub2).fit()
print (reg1.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:          NUMBEERMO_EST   R-squared:                       0.027
Model:                            OLS   Adj. R-squared:                  0.027
Method:                 Least Squares   F-statistic:                     197.7
Date:                Mon, 03 Jan 2022   Prob (F-statistic):           2.49e-44
Time:                        17:28:19   Log-Likelihood:                -38303.
No. Observations:                7226   AIC:                         7.661e+04
Df Residuals:                    7224   BIC:                         7.662e+04
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     25.5414      0.592     43.143      0.000      24.381      26.702
S2BQ1B1       31.3361      2.228     14.062      0.000      26.968      35.704
==============================================================================
```

```
Omnibus:                     7487.162   Durbin-Watson:                   2.030
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           617395.248
Skew:                           5.091   Prob(JB):                         0.00
Kurtosis:                      47.124   Cond. No.                         3.93
==============================================================================

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

10 Get the mean and Standard deviation of number of beers consumed (NUMBEERMO_EST), grouped by beer dependency (S2BQ1B1). use sub3

[9]:
```python
# hint cell 11
sub3 = sub2[['NUMBEERMO_EST', 'S2BQ1B1']].dropna()

# group means & sd
print ("Mean")
ds1 = sub3.groupby('S2BQ1B1').mean()
print (ds1)

print ("Standard deviation")
ds2 = sub3.groupby('S2BQ1B1').std()
print (ds2)
```

```
Mean
          NUMBEERMO_EST
S2BQ1B1
0.000000      25.541394
1.000000      56.877451
Standard deviation
          NUMBEERMO_EST
S2BQ1B1
0.000000      44.356007
1.000000      86.321327
```

11 Plot bar chart to show relations between number of beers consumed (NUMBEERMO_EST) and beer dependency (S2BQ1B1). use sub3

[15]:
```python
# hint cell 12
barplot = plt.figure()
ax = barplot.add_axes([0, 0, 1, 1])
ax.bar(sub3.S2BQ1B1, sub3.NUMBEERMO_EST)
```

```
ax.set_ylabel('Mean Number of Beers consumed')
ax.set_xlabel('Beer Dependence')
plt.show()
```

```
<IPython.core.display.Javascript object>
```

```
<IPython.core.display.HTML object>
```

12 Logistical Regression - Scenario 2

13 Perform Logistical Regression analysis between beer dependency (y=S2BQ1B1) and general anxiety (x=GENAXLIFE). Use sub2

[16]:
```
# hint cell 15
lreg1 = smf.logit(formula = 'S2BQ1B1 ~ GENAXLIFE', data=sub2).fit()
print (lreg1.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.212712
         Iterations 7
                           Logit Regression Results
==============================================================================
Dep. Variable:                S2BQ1B1   No. Observations:                10406
Model:                          Logit   Df Residuals:                    10404
Method:                           MLE   Df Model:                            1
Date:                Mon, 03 Jan 2022   Pseudo R-squ.:                0.007208
Time:                        17:38:01   Log-Likelihood:                -2213.5
converged:                       True   LL-Null:                       -2229.6
Covariance Type:            nonrobust   LLR p-value:                 1.434e-08
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -2.8998      0.045    -64.130      0.000      -2.988      -2.811
GENAXLIFE      0.8948      0.144      6.227      0.000       0.613       1.176
==============================================================================
```

[17]:
```
params = lreg1.params
conf = lreg1.conf_int()
conf['OR'] = params
conf.columns = ['Lower CI', 'Upper CI', 'OR']
print (np.exp(conf))
```

```
           Lower CI  Upper CI        OR
Intercept  0.050367  0.060134  0.055034
GENAXLIFE  1.846265  3.242687  2.446806
```

14 Logistical Regression - Scenario 3

[18]:
```
sub2['DYSLIFE'] = pd.to_numeric(sub2['DYSLIFE'], errors='coerce')
```

15 Perform Logistical Regression analysis between beer dependency (y=S2BQ1B1) and general anxiety (x1=GENAXLIFE) and minor depression (x2=DYSLIFE). Use sub2

[19]:
```
# hint cell 18
lreg2 = smf.logit(formula = 'S2BQ1B1 ~ GENAXLIFE + DYSLIFE', data=sub2).fit()
print (lreg2.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.212607
         Iterations 7
                           Logit Regression Results
==============================================================================
Dep. Variable:                S2BQ1B1   No. Observations:                10406
Model:                          Logit   Df Residuals:                    10403
Method:                           MLE   Df Model:                            2
Date:                Mon, 03 Jan 2022   Pseudo R-squ.:                0.007697
Time:                        17:38:47   Log-Likelihood:                -2212.4
converged:                       True   LL-Null:                       -2229.6
Covariance Type:            nonrobust   LLR p-value:                 3.524e-08
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -2.9099      0.046    -63.488      0.000      -3.000      -2.820
GENAXLIFE      0.8070      0.156      5.162      0.000       0.501       1.113
DYSLIFE        0.2638      0.174      1.513      0.130      -0.078       0.606
==============================================================================
```

[20]:
```
# 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))
```

```
           Lower CI  Upper CI        OR
Intercept  0.049798  0.059600  0.054479
GENAXLIFE  1.649620  3.044656  2.241099
DYSLIFE    0.924975  1.832270  1.301846
```

16 Logistical Regression - Scenario 4

[21]:
```
def PANIC (x1):
    if ((x1['S6Q1']==1 and x1['S6Q2']==1) or (x1['S6Q2']==1 and x1['S6Q3']==1)␣
 ↪or
    (x1['S6Q3']==1 and x1['S6Q61']==1) or (x1['S6Q61']==1 and x1['S6Q62']==1) or
    (x1['S6Q62']==1 and x1['S6Q63']==1) or (x1['S6Q63']==1 and x1['S6Q64']==1)␣
 ↪or
    (x1['S6Q64']==1 and x1['S6Q65']==1) or (x1['S6Q65']==1 and x1['S6Q66']==1)␣
 ↪or
    (x1['S6Q66']==1 and x1['S6Q67']==1) or (x1['S6Q67']==1 and x1['S6Q68']==1)␣
 ↪or
    (x1['S6Q68']==1 and x1['S6Q69']==1) or (x1['S6Q69']==1 and x1['S6Q610']==1)␣
 ↪or
    (x1['S6Q610']==1 and x1['S6Q611']==1) or (x1['S6Q611']==1 and␣
 ↪x1['S6Q612']==1) or
    (x1['S6Q612']==1 and x1['S6Q613']==1) or (x1['S6Q613']==1 and␣
 ↪x1['S6Q7']==1) or
    x1['S6Q7']==1):
        return 1
    else:
        return 0
sub2['PANIC'] = sub1.apply (lambda x1: PANIC (x1), axis=1)
c7 = sub2["PANIC"].value_counts(sort=False, dropna=False)
print(c7)
```

```
0    9596
1     921
Name: PANIC, dtype: int64
```

17 Perform Logistical Regression analysis between beer dependency (y=S2BQ1B1) and panic disorder (x=PANIC). Use sub2

[22]:
```
# hint cell 21
# logistic regression with panic
lreg3 = smf.logit(formula = 'S2BQ1B1 ~ PANIC', data = sub2).fit()
print (lreg3.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.213531
         Iterations 7
                           Logit Regression Results
==============================================================================
Dep. Variable:                S2BQ1B1   No. Observations:                10406
Model:                          Logit   Df Residuals:                    10404
Method:                           MLE   Df Model:                            1
Date:                Mon, 03 Jan 2022   Pseudo R-squ.:                0.003385
```

```
Time:                        17:39:28   Log-Likelihood:                -2222.0
converged:                       True   LL-Null:                       -2229.6
Covariance Type:            nonrobust   LLR p-value:                 0.0001023
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -2.8917      0.046    -62.875      0.000      -2.982      -2.802
PANIC          0.5210      0.127      4.102      0.000       0.272       0.770
==============================================================================
```

[ ]:
```python
# odd ratios with 95% confidence intervals
print ("Odds Ratios")
params = lreg3.params
conf = lreg3.conf_int()
conf['OR'] = params
conf.columns = ['Lower CI', 'Upper CI', 'OR']
print (np.exp(conf))
```

18 Logistical Regression - Scenario 5

19 Perform Logistical Regression analysis between beer dependency (y=S2BQ1B1) and panic disorder (x1=PANIC) and minor depression (x2=DYSLIFE). Use sub2

[23]:
```python
# hint cell 23
# logistic regression with panic and depression
lreg4 = smf.logit(formula = 'S2BQ1B1 ~ PANIC +  DYSLIFE', data = sub2).fit()
print (lreg4.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.213222
         Iterations 7
                           Logit Regression Results
==============================================================================
Dep. Variable:                S2BQ1B1   No. Observations:                10406
Model:                          Logit   Df Residuals:                    10403
Method:                           MLE   Df Model:                            2
Date:                Mon, 03 Jan 2022   Pseudo R-squ.:                0.004828
Time:                        17:40:06   Log-Likelihood:                -2218.8
converged:                       True   LL-Null:                       -2229.6
Covariance Type:            nonrobust   LLR p-value:                 2.113e-05
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -2.9109      0.047    -62.201      0.000      -3.003      -2.819
PANIC          0.4432      0.131      3.373      0.001       0.186       0.701
```

```
DYSLIFE        0.4380      0.165      2.655      0.008      0.115      0.761
==============================================================================
```

[24]:
```
# odd ratios with 95% confidence intervals
print ("Odds Ratios")
params = lreg4.params
conf = lreg4.conf_int()
conf['OR'] = params
conf.columns = ['Lower CI', 'Upper CI', 'OR']
print (np.exp(conf))
```

```
Odds Ratios
           Lower CI  Upper CI        OR
Intercept  0.049659  0.059658 0.054429
PANIC      1.203973  2.015228 1.557652
DYSLIFE    1.121488  2.141030 1.549561
```