Nhu Vo

Lab #3

Some preliminary set up code:

Show hidden output

```
from __future__ import division
import pandas as pd
import numpy as np
import statsmodels.api as sm
import statsmodels.formula.api as smf
import os
import matplotlib.pyplot as plt
```

Use the 2006 GSS again.

```
os.chdir('/Users/gregoryeirich/Desktop/Data Analysis/') # change working directory

g = pd.read_csv("GSS.2006.csv.xls")
g.head()
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-21-f824b78c1efe> in <cell line: 1>()
----> 1 os.chdir('/Users/gregoryeirich/Desktop/Data Analysis/') # change working
directory
      2
      3 g = pd.read_csv("GSS.2006.csv.xls")
      4 g.head()

FileNotFoundError: [Errno 2] No such file or directory:
'/Users/gregoryeirich/Desktop/Data Analysis/'
```

```
from google.colab import files
uploaded = files.upload()

import io
g = pd.read_csv(io.BytesIO(uploaded['GSS.2006.csv']))

g.head()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving GSS.2006.csv to GSS.2006.csv
```

	vpsu	vstrat	adults	ballot	dateintv	famgen	form	formwt	gender1	hompop	...	away7	gender14	old14	relate
0	1	1957	1	3	316	2	1	1	2	3	...	NaN	NaN	NaN	N
1	1	1957	2	2	630	1	2	1	2	2	...	NaN	NaN	NaN	N
2	1	1957	2	2	314	2	1	1	2	2	...	NaN	NaN	NaN	N
3	1	1957	1	1	313	1	2	1	2	1	...	NaN	NaN	NaN	N
4	1	1957	3	1	322	2	2	1	2	3	...	NaN	NaN	NaN	N

5 rows × 1261 columns

```
g.columns
```

```
Index(['vpsu', 'vstrat', 'adults', 'ballot', 'dateintv', 'famgen', 'form',
       'formwt', 'gender1', 'hompop',
       ...
       'away7', 'gender14', 'old14', 'relate14', 'relhh14', 'relhhd14',
       'relsp14', 'where12', 'where6', 'where7'],
      dtype='object', length=1261)
```

1. Run a simple bivariate regression, and interpret your results. (Did the results fit your expectations? Why? Why not?)

bold text

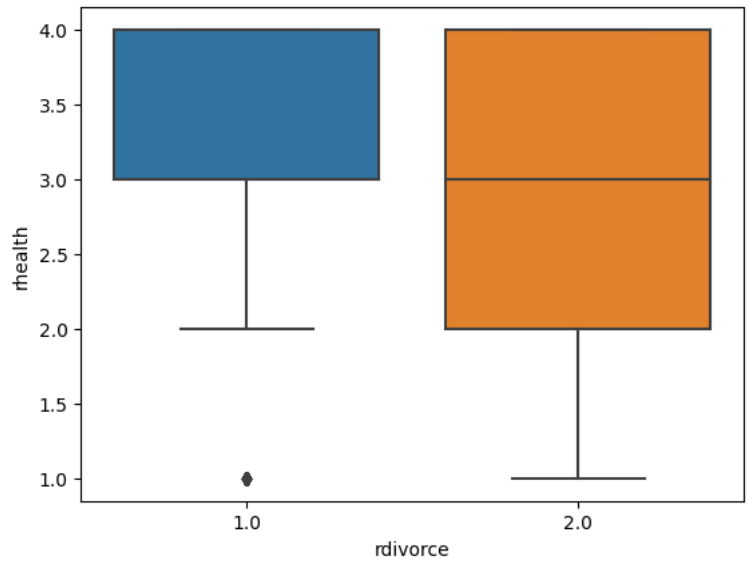
I want to see how divorced people (1=yes, 2=no) rate their own health, in general (excellent = 1, good = 2, fair = 3 or poor = 4)

```
#reverse order of "health" making higher score --> better health INSTEAD of lower score --> better health
g["rhealth"] = 5-g["health"]
```

```
#reverse order of "divorce" making 1=no divorce, 2=divorce INSTEAD of 1=divorce, 2=no divorce
g["rdivorce"] = 3-g["divorce"]
```

```
import seaborn as sns
sns.boxplot(x="rdivorce", y="rhealth",
                      data=g)
```

```
<Axes: xlabel='rdivorce', ylabel='rhealth'>
```



Looking at the boxplot above, individuals who are not divorced (1) tend to rate their health on the higher end of the scale. The results fit my expectations because I'd assume that individuals who stayed married are happier than their counterparts who had to get a divorce

2. Add an additional variable that might mediate or partly "explain" the initial association from that simple regression above -- and explain your results. Did it work out? Yes? No?

bold text

1. TRUST - Generally speaking, would you say that most people can be trusted or that you can't be too careful in life?

most people can be trusted = 1

you can't be too careful in life = 2

depends = 3

Trust could be a potential mediation because perhaps people who are divorced might be less trusting, and less trusting people are more likely to be unhappy

```
#remove "depends=3" from TRUST
g.loc[g['trust'] == 3.0, 'trust'] = np.nan
#use the loc accessor to select rows from the DataFrame g where the 'trust' column is equal to 3.0 (depends)
#The second argument, 'trust', specifies the column to be selected and replace it with nan
recode_trust = g.copy()
display(recode_trust['trust'])
```

```
0       2.0
1       2.0
2       2.0
3       NaN
4       2.0
       ...
4505    1.0
4506    2.0
4507    2.0
4508    2.0
4509    1.0
Name: trust, Length: 4510, dtype: float64
```

```
#reverse order of "health" making higher score --> better health INSTEAD of lower score --> better health
g["rhealth"] = 5-g["health"]
```

```
#reverse order of "trust" making higher score --> more trust INSTEAD of lower score --> more trust
g["rtrust"] = 3-g["trust"]
```

```
#reverse order of "divorce" making 1=no divorce, 2=divorce INSTEAD of 1=divorce, 2=no divorce
g["rdivorce"] = 3-g["divorce"]
```

```
#experimenting with method 1

recode_trust.dropna(subset=['rtrust'], inplace=True) ## We only include observations that also answer about their personality ##

lm_rep = smf.ols(formula = 'rhealth~rdivorce', data = g).fit()
print (lm_rep.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                rhealth   R-squared:                       0.006
Model:                            OLS   Adj. R-squared:                  0.006
Method:                 Least Squares   F-statistic:                     12.60
Date:                Thu, 22 Jun 2023   Prob (F-statistic):           0.000395
Time:                        01:12:53   Log-Likelihood:                -2430.4
No. Observations:                1984   AIC:                             4865.
Df Residuals:                    1982   BIC:                             4876.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      3.1772      0.056     56.423      0.000       3.067       3.288
rdivorce      -0.1500      0.042     -3.550      0.000      -0.233      -0.067
==============================================================================
```

```
Omnibus:                       81.675   Durbin-Watson:                   1.939
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               87.230
Skew:                          -0.493   Prob(JB):                     1.14e-19
Kurtosis:                       2.715   Cond. No.                         6.18
==============================================================================

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

divorce's coefficient = -0.1500: for every 1 unit of divorce increase, health decreases by -0.15, holding other variables constant. In other words, if you're divorced, you're health rating is worst. The p-value of 0 indicates that this relationship is statistically significant.

```
#experimenting with method 2
lm_reg = smf.ols(formula = 'rhealth ~ rdivorce + rtrust' , data = g).fit()
print (lm_reg.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                rhealth   R-squared:                       0.023
Model:                            OLS   Adj. R-squared:                  0.022
Method:                 Least Squares   F-statistic:                     18.25
Date:                Thu, 22 Jun 2023   Prob (F-statistic):           1.46e-08
Time:                        01:01:25   Log-Likelihood:                -1915.6
No. Observations:                1562   AIC:                             3837.
Df Residuals:                    1559   BIC:                             3853.
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      2.9654      0.088     33.691      0.000       2.793       3.138
rdivorce      -0.1777      0.047     -3.782      0.000      -0.270      -0.086
rtrust         0.1969      0.043      4.619      0.000       0.113       0.281
==============================================================================
Omnibus:                       69.321   Durbin-Watson:                   1.918
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               75.961
Skew:                          -0.526   Prob(JB):                     3.20e-17
Kurtosis:                       2.753   Cond. No.                         10.3
==============================================================================

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

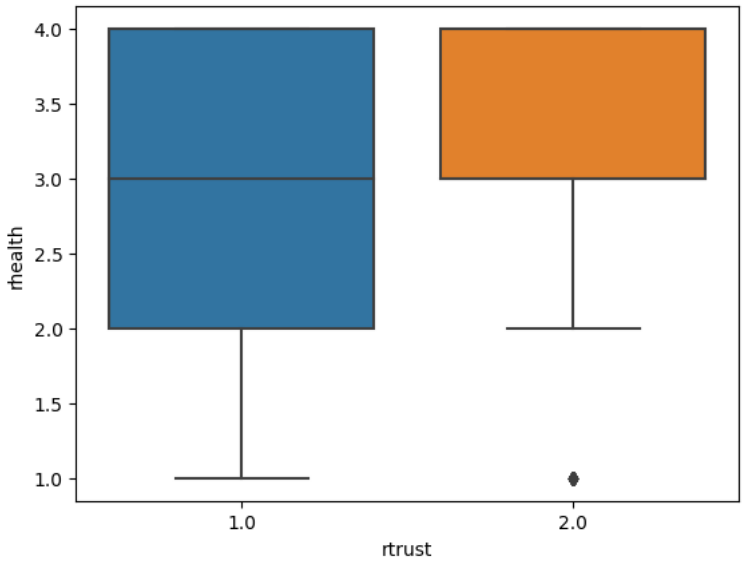
divorce's coefficient = -0.1777: For 1 unit of increase in divorce , there will be a decrease of 0.1777 units in health (holding other variables constant). This relationship is statistically significant (p-value = 0). Hence, there is a meaningful relationship between divorce and health in this model: divorced individuals have lower health.

trust coefficient = 0.1969. For 1 unit of increase in trust, there will be an increase of 0.1969 units in health (holding other variables constant). This relationship is statistically significant (p-value = 0). Hence, there is a meaningful relationship between trust and divorce in this model: individuals with higher trust have better health.

After running the multiple regression above, trust and divorce is both predictive of health.

```
sns.boxplot(x="rtrust", y="rhealth",
                      data=gg)
```

```
<Axes: xlabel='rtrust', ylabel='rhealth'>
```



Based on the visualization above, people who have higher trust in others (2) have a higher health score than their counterparts that don't (have high trust in people).

3. Run another multiple regression. Tell me how you expect your dependent variable to be affected by the independent variables. Interpret your results.

bold text

Independent variable: GOODLIFE - The way things are in America, people like me and my family have a good chance of improving our standard of living -- do you agree or disagree?

1 = Strongly agree

2 = Agree

3 = Neither agree nor disagree

4 = Disagree

5 = Strongly Disagree

Mediating Variable: SATJOB - On the whole, how satisfied are you with the work you do? Would you say you are very satisfied, moderately satisfied, a little dissatisfied or very dissatisfied?

1 = Very Satisfied

2 = Moderately Satisfied

3 = A little dissatisfied

4 = Very dissatisfied

Dependent Variable: HEALTH - Would you say your own health, in general, is excellent, good, fair or poor

1 = Excellent

2 = Good

3 = Fair

4 = Poor

```
#reverse order of health making higher score --> better instead of lower score --> better health
g["rhealth"] = 5-g["health"]
```

```
#reverse order of satjob making higher score = higher satisfaction because originally, health scale is lower score --> more satisf
g["rsatjob"] = 5-g["satjob"]
```

```
#taking "Neither agree or disagree" out from goodlife column
g.loc[g['goodlife'] == 3.0, 'goodlife'] = np.nan
#pulling location of g, pulls trust column from g,
recode_goodlife = g.copy()
display(recode_goodlife['goodlife'])
```

```
0       2.0
1       1.0
2       2.0
3       NaN
4       NaN
       ...
4505    2.0
4506    NaN
4507    NaN
4508    2.0
4509    NaN
Name: goodlife, Length: 4510, dtype: float64
```

```
#reverse order of goodlife scale, making higher score --> higher goodlife  score instead of lower goodlife score --> higher goodli
g["rgoodlife"] = 5-g["goodlife"]


lm_2 = smf.ols(formula = 'rhealth ~ rgoodlife + rsatjob' , data = g).fit()
print (lm_2.summary())

#g.dropna(subset=['jobmeans'], inplace=True)

#lm_rep = smf.ols(formula = 'health~goodlife', data = gg).fit()
#print (lm_rep.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                rhealth   R-squared:                       0.031
Model:                            OLS   Adj. R-squared:                  0.028
Method:                 Least Squares   F-statistic:                     9.776
Date:                Thu, 22 Jun 2023   Prob (F-statistic):           6.63e-05
Time:                        01:17:23   Log-Likelihood:                -670.82
No. Observations:                 606   AIC:                             1348.
Df Residuals:                     603   BIC:                             1361.
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      2.4754      0.149     16.618      0.000       2.183       2.768
rgoodlife      0.0392      0.030      1.292      0.197      -0.020       0.099
rsatjob        0.1539      0.038      4.010      0.000       0.079       0.229
==============================================================================
Omnibus:                       20.710   Durbin-Watson:                   2.021
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               22.309
Skew:                          -0.469   Prob(JB):                     1.43e-05
Kurtosis:                       2.935   Cond. No.                         23.9
==============================================================================

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

goodlife's coefficient = 0.0392: for every 1 unit of goodlife (more positive attitude about life) increase, health increases by 0.0392 units. However, this is not statistically significant (p-value = 0.197). Hence, goodlife is not predictive of health.

satjob's coefficient = 0.1539: for every 1 unit of satjob increase (the more satisfied you are at your job), health units increase by to 0.1539. This relationship is statistically significant (p-value = 0). Hence, jobsat is predictive of health: the higher the job satisfaction, the better the health.

4. Now add another independent variable to that model in Question 3, preferably a set of dummy variables. Tell me why you added that new set of variables and what effect you expected them to have. Did they have an effect? Interpret that new model.

bold text

```
#putting C in front turns it into a dummy variable
lm_3 = smf.ols(formula = 'rhealth ~ rgoodlife + rjobsat + C(race)' , data = g).fit()
print (lm_3.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                rhealth   R-squared:                       0.054
Model:                            OLS   Adj. R-squared:                  0.048
Method:                 Least Squares   F-statistic:                     8.571
Date:                Thu, 22 Jun 2023   Prob (F-statistic):           9.91e-07
Time:                        01:07:41   Log-Likelihood:                -663.68
No. Observations:                 606   AIC:                             1337.
Df Residuals:                     601   BIC:                             1359.
Df Model:                           4
Covariance Type:            nonrobust
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
Intercept        2.5057      0.149     16.847      0.000       2.214       2.798
C(race)[T.2]    -0.0622      0.092     -0.676      0.499      -0.243       0.118
C(race)[T.3]    -0.3106      0.082     -3.785      0.000      -0.472      -0.149
rgoodlife        0.0519      0.030      1.717      0.087      -0.007       0.111
rjobsat          0.1507      0.038      3.963      0.000       0.076       0.225
==============================================================================
Omnibus:                       21.937   Durbin-Watson:                   2.041
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               23.614
Skew:                          -0.483   Prob(JB):                     7.45e-06
Kurtosis:                       3.032   Cond. No.                         24.2
==============================================================================

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

note:

white = 1

black = 2

other = 3

Interpretation: net/regardless of goodlife and satjob,

• relative to whites, blacks "C(race)[T.2]" have health that is 0.0622 lower (coef=-0.0622)

• relative to whites, those of other race "C(race)[T.3]" have health that is 0.3106 lower (coef=-0.3106)

• for a white person with 0 goodlife and 0 jobsat, they'd have 2.5057 health score out of 4 (intercept = 2.5057)