Assignment 4 REAL

January 22, 2023

0.0.1 1. Reading in our libraries, out dataset, and renaming our variables

[1]:
```
!pip install researchpy
```

```
Requirement already satisfied: researchpy in /opt/conda/lib/python3.9/site-
packages (0.3.5)
Requirement already satisfied: pandas in /opt/conda/lib/python3.9/site-packages
(from researchpy) (1.3.5)
Requirement already satisfied: statsmodels in /opt/conda/lib/python3.9/site-
packages (from researchpy) (0.12.2)
Requirement already satisfied: numpy in /opt/conda/lib/python3.9/site-packages
(from researchpy) (1.21.6)
Requirement already satisfied: scipy in /opt/conda/lib/python3.9/site-packages
(from researchpy) (1.7.3)
Requirement already satisfied: patsy in /opt/conda/lib/python3.9/site-packages
(from researchpy) (0.5.3)
Requirement already satisfied: python-dateutil>=2.7.3 in
/opt/conda/lib/python3.9/site-packages (from pandas->researchpy) (2.8.0)
Requirement already satisfied: pytz>=2017.3 in /opt/conda/lib/python3.9/site-
packages (from pandas->researchpy) (2021.1)
Requirement already satisfied: six in /opt/conda/lib/python3.9/site-packages
(from patsy->researchpy) (1.16.0)
```

[2]:
```
# First, We're going to call in our libraries
from IPython.display import Image
import researchpy as rp
import numpy as np
import pandas as pd
import math
from scipy import stats
from scipy.stats import ttest_ind, chi2_contingency
import seaborn as sns
import matplotlib as plt
import matplotlib.pyplot as plt
import scipy


pd.options.display.float_format = '{:.4f}'.format
```

[3]:
```
%matplotlib inline
```

[4]:
```
import warnings
warnings.filterwarnings("ignore")
```

[5]:
```
chis_df = pd.read_csv('CHISextract2022.csv')
chis_df
```

[5]:

	AB1	AJ105	AK23	AK25	AM19	AM20	AM21	AK28	AM45	AM48	…	\
0	2	-1	1	1	3	4	3	3	4	2	…	
1	2	-1	3	2	3	3	3	2	2	2	…	
2	2	-1	1	2	2	3	2	1	5	2	…	
3	3	1	1	1	3	3	1	2	1	2	…	
4	3	-1	1	1	1	4	1	2	5	2	…	
…	…	…	…	…	…	…	…	…	…	…	…	
24448	2	-1	3	2	3	1	2	1	3	2	…	
24449	4	1	3	2	2	2	2	2	4	2	…	
24450	4	-1	3	2	2	2	3	3	3	1	…	
24451	4	1	3	2	2	2	2	3	5	2	…	
24452	3	-1	3	2	2	3	2	1	3	2	…	

	OCCMAIN2	AHEDC_P1	AK22_P1	AK3_P1V2	HHSIZE_P1	OMBSRR_P1	RACECN_P1	\
0	-1	4	2	-1	3	1	1	
1	99	8	8	7	1	1	1	
2	7	7	7	2	1	2	5	
3	-1	2	6	-1	6	1	1	
4	10	6	9	6	1	2	5	
…	…	…	…	…	…	…	…	
24448	5	3	1	6	1	3	4	
24449	5	1	3	6	2	1	5	
24450	-1	9	12	-1	2	3	4	
24451	5	3	1	2	2	1	5	
24452	1	4	2	4	3	1	5	

	SRAGE_P1	TIMEAD_P1	TIMENEV2_P1
0	55	17	-1
1	30	13	13
2	65	13	13
3	55	18	-1
4	55	13	13
…	…	…	…
24448	30	15	-1
24449	40	17	-1
24450	60	2	2
24451	60	13	13
24452	18	13	-1

```
[24453 rows x 53 columns]
```

[6]:
```
chis_df.rename(columns={"AM19":"help",
                        "AH33NEW":"immigrant",
                        "OMBSRR_P1": "race_ethnicity",
                        "AM3" : "cut_meal",
                       "CVA3":"wit_crime"}, inplace=True)
```

[7]:
```
chis_df_small=(chis_df[['help', 'race_ethnicity', 'immigrant', 'cut_meal',␣
↪'wit_crime',]])
```

[8]:
```
chis_df_small
```

[8]:

	help	race_ethnicity	immigrant	cut_meal	wit_crime
0	3	1	1	2	-1
1	3	1	1	-1	-1
2	2	2	1	-1	-1
3	3	1	2	-1	-1
4	1	2	1	-1	-1
…	…	…	…	…	…
24448	3	3	1	2	-1
24449	2	1	2	1	-1
24450	2	3	1	-1	-1
24451	2	1	2	2	-1
24452	2	1	1	2	-1

```
[24453 rows x 5 columns]
```

1 Codebook

AM19: People in Neighborhood Willing to Help Each Other (-2=Proxy Skipped, 1=Strongly Agree, 2=Agree, 3=Disagree, 4=Strongly Disagree)

SRSEX: Self-reported Sex (1= Male, 2=Female)

OMBSRR_P1: Race/ethnicity (1=Hispanic, 2= White NH, 3=Black NH, 4=AmIndian/Alaska Native NH, 5=Asian NH, 6=Other or two or more)

POVLL: poverty level (1 = 0-99% FPL, 2=100-199% FPL, 3=200-299% FPL, 4=300% FPL and above)

AK22_P1: Household Income

AH33NEW: Born in/outside the U.S. (1 = Born in the U.S., 2 = Born outside the U.S.)

AM3: All Adults Cut/Skip meals in the past 12 months for money (-2 = proxy skipped, -1 = n/a, 1 = yes, 2 = no)

CVA3: Witnessed Another AAPI Person Treated Unfairly Due to Race/Ethnicity (-1=Inapplicable, 1=Yes, 2=No)

1.0.1 2. Cleaning Variables

Asian

[9]:
```
chis_df_1=pd.get_dummies(chis_df_small, columns=['race_ethnicity'])
chis_df_1
```

[9]:
```
       help  immigrant  cut_meal  wit_crime  race_ethnicity_1  \
0         3          1         2         -1                 1
1         3          1        -1         -1                 1
2         2          1        -1         -1                 0
3         3          2        -1         -1                 1
4         1          1        -1         -1                 0
…     …         …        …         …                  …
24448     3          1         2         -1                 0
24449     2          2         1         -1                 1
24450     2          1        -1         -1                 0
24451     2          2         2         -1                 1
24452     2          1         2         -1                 1

       race_ethnicity_2  race_ethnicity_3  race_ethnicity_4  race_ethnicity_5  \
0                     0                 0                 0                 0
1                     0                 0                 0                 0
2                     1                 0                 0                 0
3                     0                 0                 0                 0
4                     1                 0                 0                 0
…                 …                 …                 …                 …
24448                 0                 1                 0                 0
24449                 0                 0                 0                 0
24450                 0                 1                 0                 0
24451                 0                 0                 0                 0
24452                 0                 0                 0                 0

       race_ethnicity_6
0                     0
1                     0
2                     0
3                     0
4                     0
…                 …
24448                 0
24449                 0
24450                 0
24451                 0
24452                 0

[24453 rows x 10 columns]
```

[10]:
```
pd.crosstab(chis_df_1['race_ethnicity_5'], columns='Total')
```

[10]:
```
col_0              Total
race_ethnicity_5
0                  20491
1                   3962
```

[7]:
```
chis_df_small['race_ethnicity_dv']=chis_df_small['race_ethnicity_5']
pd.crosstab(chis_df_small['race_ethnicity_dv'], columns='count')
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In [7], line 1
----> 1 chis_df_small['race_ethnicity_dv']=chis_df_small['race_ethnicity_5']
      2 pd.crosstab(chis_df_small['race_ethnicity_dv'], columns='count')

NameError: name 'chis_df_small' is not defined
```

Cut meals

[11]:
```
chis_df_1=pd.get_dummies(chis_df_small, columns=['cut_meal'])
chis_df_1
```

[11]:

	help	race_ethnicity	immigrant	wit_crime	cut_meal_-2	cut_meal_-1	\
0	3	1	1	-1	0	0	
1	3	1	1	-1	0	1	
2	2	2	1	-1	0	1	
3	3	1	2	-1	0	1	
4	1	2	1	-1	0	1	
…	…	…	…	…	…	…	
24448	3	3	1	-1	0	0	
24449	2	1	2	-1	0	0	
24450	2	3	1	-1	0	1	
24451	2	1	2	-1	0	0	
24452	2	1	1	-1	0	0	

	cut_meal_1	cut_meal_2
0	0	1
1	0	0
2	0	0
3	0	0
4	0	0
…	…	…
24448	0	1
24449	1	0
24450	0	0
24451	0	1
24452	0	1

```
[24453 rows x 8 columns]
```

[12]:
```
pd.crosstab(chis_df_1['cut_meal_1'], columns='Total')
```

[12]:
```
col_0       Total
cut_meal_1
0           23347
1            1106
```

[13]:
```
pd.crosstab(chis_df_small['cut_meal'], columns='Total')
```

[13]:
```
col_0     Total
cut_meal
-2            5
-1        18814
 1         1106
 2         4528
```

[14]:
```
chis_df_small['cut_meal_dv']=chis_df_small['cut_meal'].map({2:0, 1:1, -2:0, -1:
↪0})
pd.crosstab(chis_df_small['cut_meal_dv'], columns='count')
```

[14]:
```
col_0        count
cut_meal_dv
0            23347
1             1106
```

Immigrant

[15]:
```
chis_df_1=pd.get_dummies(chis_df_small, columns=['immigrant'])
chis_df_1
```

[15]:
```
       help  race_ethnicity  cut_meal  wit_crime  cut_meal_dv  immigrant_1  \
0         3               1         2         -1            0            1
1         3               1        -1         -1            0            1
2         2               2        -1         -1            0            1
3         3               1        -1         -1            0            0
4         1               2        -1         -1            0            1
…     …             …        …         …           …           …
24448     3               3         2         -1            0            1
24449     2               1         1         -1            1            0
24450     2               3        -1         -1            0            1
24451     2               1         2         -1            0            0
24452     2               1         2         -1            0            1

       immigrant_2
0                0
```

```
1                0
2                0
3                1
4                0
…              …
24448            0
24449            1
24450            0
24451            1
24452            0

[24453 rows x 7 columns]
```

[16]:
```
pd.crosstab(chis_df_1['immigrant_2'], columns='Total')
```

[16]:
```
col_0         Total
immigrant_2
0             18122
1              6331
```

[17]:
```
pd.crosstab(chis_df_small['immigrant'], columns='Total')
```

[17]:
```
col_0       Total
immigrant
1           18122
2            6331
```

[18]:
```
chis_df_small['immigrant_dv']=chis_df_small['immigrant'].map({2:1, 1:0})
pd.crosstab(chis_df_small['immigrant_dv'], columns='count')
```

[18]:
```
col_0          count
immigrant_dv
0              18122
1               6331
```

Help

[19]:
```
pd.crosstab(chis_df_small['help'], columns='count')
```

[19]:
```
col_0  count
help
-2        12
 1      5601
 2     15074
 3      3125
 4       641
```

[20]:
```
#need to insert code here that drops -2, which are the non-respondants

chis_df_1 = chis_df_small[chis_df_small['help'] > 0]
chis_df_1['help'].describe()
```

[20]:
```
count   24441.0000
mean        1.9511
std         0.6779
min         1.0000
25%         2.0000
50%         2.0000
75%         2.0000
max         4.0000
Name: help, dtype: float64
```

[21]:
```
#Combined -2, 1, and 2 as 1, and then combined 3 and 4 as 0

chis_df_small['help_dv']=chis_df_small['help'].map({-2:1, 1:1, 2:1, 3:0, 4:0})
pd.crosstab(chis_df_small['help_dv'], columns='count')
```

[21]:
```
col_0     count
help_dv
0          3766
1         20687
```

Witness Crime

[22]:
```
#here I created a new data frame where I dropped the -1, which is the number of␣
 ↪nonrespondants
chis_df_1 = chis_df_small[chis_df_small['wit_crime'] > 0]
chis_df_1['wit_crime'].describe()
```

[22]:
```
count   4441.0000
mean       1.8151
std        0.3882
min        1.0000
25%        2.0000
50%        2.0000
75%        2.0000
max        2.0000
Name: wit_crime, dtype: float64
```

[23]:
```
chis_df_1.describe()
```

[23]:
```
            help  race_ethnicity  immigrant  cut_meal  wit_crime  cut_meal_dv  \
count 4441.0000       4441.0000  4441.0000 4441.0000  4441.0000    4441.0000
mean     1.9912          4.9768     1.6620   -0.3986     1.8151       0.0290
```

```
std      0.6030          0.7496     0.4731    1.1775     0.3882       0.1680
min     -2.0000          1.0000     1.0000   -2.0000     1.0000       0.0000
25%      2.0000          5.0000     1.0000   -1.0000     2.0000       0.0000
50%      2.0000          5.0000     2.0000   -1.0000     2.0000       0.0000
75%      2.0000          5.0000     2.0000   -1.0000     2.0000       0.0000
max      4.0000          6.0000     2.0000    2.0000     2.0000       1.0000

       immigrant_dv   help_dv
count     4441.0000 4441.0000
mean         0.6620    0.8581
std          0.4731    0.3489
min          0.0000    0.0000
25%          0.0000    1.0000
50%          1.0000    1.0000
75%          1.0000    1.0000
max          1.0000    1.0000
```

[24]:
```
#here we dropped -1 which is
pd.crosstab(chis_df_1['wit_crime'], columns='count')
```

[24]:
```
col_0       count
wit_crime
1             821
2            3620
```

[25]:
```
##need to drop inapplicable row which is -1, here I created a dummy variable␣
 ↪where 1 is yes, and 0 is no

chis_df_1['wit_crime_dv']=chis_df_1['wit_crime'].map({1:1, 2:0})
pd.crosstab(chis_df_1['wit_crime_dv'], columns='count')
```

[25]:
```
col_0          count
wit_crime_dv
0               3620
1                821
```

Meals

1.0.2 3. Testing Bivariate Relationships

Help and Crime

[26]:
```
#let's look at whether there is any relationship between inclination to help a␣
 ↪neighbor and witnessins AAPI hate
pd.crosstab(index=chis_df_1["wit_crime_dv"], columns=chis_df_1["help_dv"],␣
 ↪margins=True, normalize='index')
```

[26]:
```
help_dv           0      1
wit_crime_dv
0            0.1279 0.8721
1            0.2034 0.7966
All          0.1419 0.8581
```

[27]:
```
chis_df_1['wit_crime_dv']=chis_df_1['wit_crime'].map({1:1, 2:0})
pd.crosstab(chis_df_1['wit_crime_dv'], columns='count')
```

[27]:
```
col_0         count
wit_crime_dv
0              3620
1               821
```

Help and Immigrant

[28]:
```
#let's look at whether there is any relationship between inclination to help a␣
 ↪neighbor and Immigrant status
pd.crosstab(index=chis_df_1["immigrant_dv"], columns=chis_df_1["help_dv"],␣
 ↪margins=True, normalize='index')
```

[28]:
```
help_dv           0      1
immigrant_dv
0            0.1392 0.8608
1            0.1432 0.8568
All          0.1419 0.8581
```

[29]:
```
chis_df_1['immigrant_dv']=chis_df_1['immigrant'].map({1:1, 2:0})
pd.crosstab(chis_df_1['immigrant_dv'], columns='count')
```

[29]:
```
col_0         count
immigrant_dv
0              2940
1              1501
```

Help and cut meals

[30]:
```
#let's look at whether there is any relationship between inclination to help a␣
 ↪neighbor and Immigrant status
pd.crosstab(index=chis_df_1["cut_meal_dv"], columns=chis_df_1["help_dv"],␣
 ↪margins=True, normalize='index')
```

[30]:
```
help_dv           0      1
cut_meal_dv
0            0.1378 0.8622
1            0.2791 0.7209
All          0.1419 0.8581
```

[31]:
```
chis_df_1['cut_meal_dv']=chis_df_1['cut_meal'].map({1:1, 2:0, -1:0, -2:0})
pd.crosstab(chis_df_1['cut_meal_dv'], columns='count')
```

[31]:
```
col_0        count
cut_meal_dv
0             4312
1              129
```

Asian and Crime

[1]:
```
#let's look at whether there is any relationship between Asian and witnessins␣
 ↪AAPI hate
pd.crosstab(index=chis_df_1["wit_crime_dv"],␣
 ↪columns=chis_df_1["race_ethnicity_5"], margins=True, normalize='index')
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In [1], line 2
      1 #let's look at whether there is any relationship between Asian and␣
 ↪witnessins AAPI hate
----> 2 pd.crosstab(index=chis_df_1["wit_crime_dv"],␣
 ↪columns=chis_df_1["race_ethnicity_5"], margins=True, normalize='index')

NameError: name 'pd' is not defined
```

[ ]:

1.0.3 4. Chi-square test

[32]:
```
#let's check first and make sure we have at least 5 observations in each cell
pd.crosstab(index=chis_df_1["help_dv"], columns=chis_df_1["wit_crime_dv"],␣
 ↪margins=True)
```

[32]:

wit_crime_dv	0	1	All
help_dv			
0	463	167	630
1	3157	654	3811
All	3620	821	4441

[33]:
```
pd.crosstab(index=chis_df_1["help_dv"], columns=chis_df_1["wit_crime_dv"],␣
 ↪margins=True, normalize='columns')
```

[33]:

wit_crime_dv	0	1	All
help_dv			
0	0.1279	0.2034	0.1419
1	0.8721	0.7966	0.8581

1.0.4 4.1 Chi-square Help and Crime

[34]:
```
#here's researchpy again, this time for the chi-square test
rp.crosstab(chis_df_1["wit_crime_dv"], chis_df_1["help_dv"], prop="col",␣
 ↪test="chi-square")
```

[34]:
```
(              help_dv
 help_dv              0        1      All
 wit_crime_dv
 0              73.4900  82.8400  81.5100
 1              26.5100  17.1600  18.4900
 All           100.0000 100.0000 100.0000,
                  Chi-square test  results
 0  Pearson Chi-square ( 1.0) =   31.3445
 1                     p-value =    0.0000
 2                Cramer's phi =    0.0840)
```

1.0.5 4.2 Chi-square Help and Immigrant

[35]:
```
pd.crosstab(index=chis_df_1["help_dv"], columns=chis_df_1["immigrant_dv"],␣
 ↪margins=True)
```

[35]:
```
immigrant_dv     0     1   All
help_dv
0              421   209   630
1             2519  1292  3811
All           2940  1501  4441
```

[36]:
```
pd.crosstab(index=chis_df_1["help_dv"], columns=chis_df_1["immigrant_dv"],␣
 ↪margins=True, normalize='columns')
```

[36]:
```
immigrant_dv      0      1    All
help_dv
0            0.1432 0.1392 0.1419
1            0.8568 0.8608 0.8581
```

[37]:
```
#here's researchpy again, this time for the chi-square test
rp.crosstab(chis_df_1["immigrant_dv"], chis_df_1["help_dv"], prop="col",␣
 ↪test="chi-square")
```

[37]:
```
(              help_dv
 help_dv              0        1      All
 immigrant_dv
 0              66.8300  66.1000  66.2000
 1              33.1700  33.9000  33.8000
 All           100.0000 100.0000 100.0000,
                  Chi-square test  results
```

```
0  Pearson Chi-square ( 1.0) =    0.1278
1                     p-value =    0.7207
2                Cramer's phi =    0.0054)
```

1.0.6 4.3 Chi-square Help and Meals Cut

[38]:
```
pd.crosstab(index=chis_df_1["help_dv"], columns=chis_df_1["cut_meal_dv"],␣
↪margins=True)
```

[38]:
```
cut_meal_dv     0    1   All
help_dv
0              594   36   630
1             3718   93  3811
All           4312  129  4441
```

[39]:
```
pd.crosstab(index=chis_df_1["help_dv"], columns=chis_df_1["cut_meal_dv"],␣
↪margins=True, normalize='columns')
```

[39]:
```
cut_meal_dv      0      1    All
help_dv
0           0.1378 0.2791 0.1419
1           0.8622 0.7209 0.8581
```

[40]:
```
rp.crosstab(chis_df_1["cut_meal_dv"], chis_df_1["help_dv"], prop="col",␣
↪test="chi-square")
```

[40]:
```
(             help_dv
 help_dv            0        1      All
 cut_meal_dv
 0            94.2900  97.5600  97.1000
 1             5.7100   2.4400   2.9000
 All         100.0000 100.0000 100.0000,
                 Chi-square test  results
 0  Pearson Chi-square ( 1.0) =   20.5468
 1                     p-value =    0.0000
 2                Cramer's phi =    0.0680)
```