Practical 1 - JosiahTeh

December 8, 2021

1 First Name: Josiah

2 Last Name: Teh

3 Importing libraries

[1]:
```
#import pandas & numpy
import pandas as pd
import numpy as np
```

4 1. Read in the nesarc.csv file

[2]:
```
#read in csv file into
nesarc = pd.read_csv('nesarc.csv', low_memory=False) #increase efficiency
```

5 2. Print the number of rows, columns in nesarc

[3]:
```
# hint lecture cell 3
print (len(nesarc)) #number of rows (observations)
print (len(nesarc.columns)) # number of columns (variables)
```

```
43093
760
```

6 Printing the first 5 rows of nesarc

[8]:
```
# hint lecture cell 3
nesarc.head(5)
```

[8]:

	S3BQ1A1	ETHRACE2A	ETOTLCA2	IDNUM	PSU	STRATUM	WEIGHT	CDAY	\
0	0	5		1	4007	403	3928.613505	14	
1	1	5	0.0014	2	6045	604	3638.691845	12	
2	2	5		3	12042	1218	5779.032025	23	
3	3	5		4	17099	1704	1071.754303	9	
4	4	2		5	17099	1704	4986.952377	18	

```
   CMON  CYEAR  …  SOLP12ABDEP  HAL12ABDEP  HALP12ABDEP  MAR12ABDEP  \
0     8   2001  …            0           0            0           0
1     1   2002  …            0           0            0           0
2    11   2001  …            0           0            0           0
3     9   2001  …            0           0            0           0
4    10   2001  …            0           0            0           0

   MARP12ABDEP  HER12ABDEP  HERP12ABDEP  OTHB12ABDEP  OTHBP12ABDEP  NDSymptoms
0            0           0            0            0             0         NaN
1            0           0            0            0             0         NaN
2            0           0            0            0             0         NaN
3            0           0            0            0             0         NaN
4            0           0            0            0             0         NaN

[5 rows x 760 columns]
```

7 Convert Alcohol effects - 12 months (S2BQ1B1) to numeric & print first 10 rows

[9]:
```
# hint lecture cell 4
#Read in Alcohol effects - 12 months (S2BQ1B1)
nesarc['S2BQ1B1'] = pd.to_numeric(nesarc['S2BQ1B1'], errors='coerce')#convert␣
 ↪variable to numeric
#print the first 10 rows
nesarc['S2BQ1B1'].head(10)
```

[9]:
```
0    NaN
1    2.0
2    NaN
3    NaN
4    NaN
5    2.0
6    2.0
7    2.0
8    2.0
9    1.0
Name: S2BQ1B1, dtype: float64
```

8 Print the count and percentage of Alcohol effects - 12 months (S2BQ1B1)

[12]:
```
# hint lecture cell 9
#calculate counts for Alcohol effects - 12 months (S2BQ1B1)
print ('counts for S2BQ1B1 alcohol effect in the past 12 months, yes=1')␣
 ↪#better titles
c_al_dep = nesarc['S2BQ1B1'].value_counts(sort=False)#sort by values (not count)
print (c_al_dep)

#calculate percentages for Alcohol effects - 12 months (S2BQ1B1)
print ('percentages for S2BQ1B1 alcohol effect in the past 12 months, yes=1')␣
 ↪#better titles
p_al_dep = nesarc['S2BQ1B1'].value_counts(sort=False, normalize=True)␣
 ↪#normalize=True will give percentage
print (p_al_dep)
```

```
counts for S2BQ1B1 alcohol effect in the past 12 months, yes=1
2.0    25309
9.0      311
1.0     1326
Name: S2BQ1B1, dtype: int64
percentages for S2BQ1B1 alcohol effect in the past 12 months, yes=1
2.0    0.939249
9.0    0.011542
1.0    0.049210
Name: S2BQ1B1, dtype: float64
```

9 Convert Beer drinking status (S2AQ5A) to numeric & print first 10 rows

[15]:
```
# hint lecture cell 10
nesarc['S2AQ5A'] = pd.to_numeric(nesarc['S2AQ5A'], errors='coerce') #convert␣
 ↪smoking status to numeric
nesarc['S2AQ5A'].head(10) #print the first 25
```

[15]:
```
0    NaN
1    1.0
2    NaN
3    NaN
4    NaN
5    2.0
6    2.0
7    2.0
8    1.0
9    2.0
```

```
Name: S2AQ5A, dtype: float64
```

10 Print the count and percentage of Beer drinking status (S2AQ5A)

[21]:
```
# hint lecture cell 11
c_beer_status = nesarc['S2AQ5A'].value_counts(sort=False, dropna=False)␣
 ↪#dropna=False to keep NaN in calculation
print ('counts for S2AQ5A beer drinking in the past year, yes=1')
print(c_beer_status)

p_beer_status = nesarc['S2AQ5A'].value_counts(sort=False, normalize=True,␣
 ↪dropna=False)
print ('percentages for S2AQ5A beer drinking in the past year, yes=1')
print (p_beer_status)
```

```
counts for S2AQ5A beer drinking in the past year, yes=1
NaN    16147
2.0     8562
9.0       38
1.0    18346
Name: S2AQ5A, dtype: int64
percentages for S2AQ5A beer drinking in the past year, yes=1
NaN    0.374701
2.0    0.198687
9.0    0.000882
1.0    0.425730
Name: S2AQ5A, dtype: float64
```

11 Convert HOW OFTEN DRANK BEER IN LAST 12 MONTHS (S2AQ5B) to numeric & print first 10 rows

[22]:
```
# hint lecture cell 10
nesarc['S2AQ5B'] = pd.to_numeric(nesarc['S2AQ5B'], errors='coerce')
nesarc['S2AQ5B'].head(10)
```

[22]:
```
0     NaN
1    10.0
2     NaN
3     NaN
4     NaN
5     NaN
6     NaN
7     NaN
8     9.0
```

```
9     NaN
Name: S2AQ5B, dtype: float64
```

12 Print the count and percentage of HOW OFTEN DRANK BEER IN LAST 12 MONTHS (S2AQ5B)

[23]:
```
# hint lecture cell 12
nesarc['S2AQ5B'] = nesarc['S2AQ5B'].astype('category') #set the data type as␣
 ↪categorical data

c_beer_feq = nesarc['S2AQ5B'].value_counts(sort=False, dropna=False)
print ('counts for S2AQ5B - usual frequency when drinking beer')
print(c_beer_feq)

p_beer_feq = nesarc['S2AQ5B'].value_counts(sort=False, dropna=False,␣
 ↪normalize=True)
print ('percentages for S2AQ5B - usual frequency when drinking beer')
print (p_beer_feq)
```

```
counts for S2AQ5B - usual frequency when drinking beer
1.0       836
2.0       645
3.0      1535
4.0      2190
5.0      2451
6.0      2603
7.0      2127
8.0      1194
9.0      2268
10.0     2442
99.0       55
NaN     24747
Name: S2AQ5B, dtype: int64
percentages for S2AQ5B - usual frequency when drinking beer
1.0     0.019400
2.0     0.014968
3.0     0.035621
4.0     0.050820
5.0     0.056877
6.0     0.060404
7.0     0.049358
8.0     0.027708
9.0     0.052630
10.0    0.056668
99.0    0.001276
NaN     0.574270
```

```
Name: S2AQ5B, dtype: float64
```

13 Convert NUMBER OF BEERS USUALLY CONSUMED ON DAYS WHEN DRANK BEER IN LAST 12 MONTHS (S2AQ5D) to numeric & print first 10 rows

[24]:
```
# hint lecture cell 10
nesarc['S2AQ5D'] = pd.to_numeric(nesarc['S2AQ5D'], errors = 'coerce')
nesarc['S2AQ5D'].head(10)
```

[24]:
```
0    NaN
1    1.0
2    NaN
3    NaN
4    NaN
5    NaN
6    NaN
7    NaN
8    1.0
9    NaN
Name: S2AQ5D, dtype: float64
```

14 Print the count and percentage of NUMBER OF BEERS USUALLY CONSUMED ON DAYS WHEN DRANK BEER IN LAST 12 MONTHS (S2AQ5D)

[25]:
```
# hint lecture cell 11
nesarc['S2AQ5D'] = nesarc['S2AQ5D'].astype('category')
c_beer_quan = nesarc['S2AQ5D'].value_counts(sort=False, dropna=False)
print ('counts for S2AQ5D usual quantity when drink beer')
print(c_beer_quan)

p_beer_quan = nesarc['S2AQ5D'].value_counts(sort=False, dropna=False,␣
 ↪normalize=True)
print ('percentages for S2AQ5D usual quantity when drink beer')
print (p_beer_quan)
```

```
counts for S2AQ5D usual quantity when drink beer
1.0      7122
2.0      4938
3.0      2564
4.0      1224
5.0       507
6.0      1128
7.0       118
```

```
8.0       205
9.0        28
10.0      108
11.0        6
12.0      231
13.0        3
14.0        6
15.0       21
16.0        1
17.0        4
18.0       18
20.0        7
24.0       23
25.0        1
30.0        3
36.0        1
42.0        1
99.0       78
NaN     24747
Name: S2AQ5D, dtype: int64
percentages for S2AQ5D usual quantity when drink beer
1.0     0.165270
2.0     0.114589
3.0     0.059499
4.0     0.028404
5.0     0.011765
6.0     0.026176
7.0     0.002738
8.0     0.004757
9.0     0.000650
10.0    0.002506
11.0    0.000139
12.0    0.005360
13.0    0.000070
14.0    0.000139
15.0    0.000487
16.0    0.000023
17.0    0.000093
18.0    0.000418
20.0    0.000162
24.0    0.000534
25.0    0.000023
30.0    0.000070
36.0    0.000023
42.0    0.000023
99.0    0.001810
NaN     0.574270
Name: S2AQ5D, dtype: float64
```

15 Use groupby () to calculate count & percentage for Alcohol effects - 12 months (S2BQ1B1)

[26]:
```
# hint lecture cell 14
#count using groupby
c_al_dep_alt = nesarc.groupby("S2BQ1B1").size()
print(c_al_dep_alt)
```

```
S2BQ1B1
1.0     1326
2.0    25309
9.0      311
dtype: int64
```

[27]:
```
# hint lecture cell 15
p_al_dep_alt = nesarc.groupby('S2BQ1B1').size()*100/len(nesarc)
print(p_al_dep_alt)
```

```
S2BQ1B1
1.0     3.077066
2.0    58.731116
9.0     0.721695
dtype: float64
```

16 Obtain a subset of nesarc data, with the following criteria

17 Age from 26 to 50

18 Beer drinking status - S2AQ5A = Y

[28]:
```
# hint lecture cell 16
nesarc['AGE'] = pd.to_numeric(nesarc['AGE'], errors='coerce')

#subset data to young adults age 26 to 50 who have drink beer in the past 12␣
 ↪months
sub1= nesarc[(nesarc['AGE']>=26) & (nesarc['AGE']<=50) & (nesarc['S2AQ5A']==1)]

#make a copy of the new subsetted data
sub2 = sub1.copy()

c5 = sub2['AGE'].value_counts(sort=False)
print ('counts for AGE')
print(c5)

p5 = sub2['AGE'].value_counts(sort=False, normalize=True)
print ('percentages for AGE')
print (p5)
```

```
counts for AGE
32    502
40    497
48    377
33    423
41    445
49    331
26    325
34    462
42    463
50    325
27    397
35    416
43    398
28    347
36    464
44    381
29    407
37    498
45    434
30    443
38    504
46    396
31    453
39    464
47    365
Name: AGE, dtype: int64
percentages for AGE
32    0.047732
40    0.047257
```

```
48    0.035847
33    0.040221
41    0.042312
49    0.031473
26    0.030902
34    0.043929
42    0.044024
50    0.030902
27    0.037748
35    0.039555
43    0.037843
28    0.032994
36    0.044119
44    0.036227
29    0.038699
37    0.047352
45    0.041267
30    0.042122
38    0.047922
46    0.037653
31    0.043073
39    0.044119
47    0.034706
Name: AGE, dtype: float64
```