Importing libraries

In [1]:

```
#import pandas & numpy
import pandas as pd
import numpy as np
```

1. Read in the nesarc.csv file

In [2]:

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

2. Print the number of rows, columns in nesarc

In [3]:

```
print (len(nesarc)) #number of rows (observations)
print (len(nesarc.columns)) # number of columns (variables)
```

```
43093
3010
```

There are 43093 rows and 3010 columns in the DataFrame.

Printing the first 5 rows of nesarc

In [4]:

```
nesarc.head() #print the first five rows
```

Out[4]:

The first 5 rows of DataFrame.

	Unnamed: 0	ETHRACE2A	ETOTLCA2	IDNUM	PSU	STRATUM	WEIGHT	C
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

5 rows × 3010 columns

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

In [5]:

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

Out[5]:

```
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
```

The first 10 rows of 12 months alcohol effects, 'NaN' means the cell is empty or has invalid input.

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

In [6]:

```
#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 t
itles
p_al_dep = nesarc['S2BQ1B1'].value_counts(sort=False, normalize=True) #normalize=True w
ill give percentage
print (p_al_dep)
```

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

Due to alcohol effects, in last 12 months, 25309 interviewees had 2 abuses (93.92% of total sample), 1326 had 1 abuses (4.92% of total sample), and 311 had 9 abuses (1.15% of total sample).

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

In [7]:

```
nesarc['S2AQ5A'] = pd.to_numeric(nesarc['S2AQ5A'], errors='coerce') #convert smoking st
atus to numeric
nesarc['S2AQ5A'].head(10) #print the first 25
```

Out[7]:

```
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
```

First 10 rows of data of any beer consumption in last 12 months.

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

In [8]:

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

In last 12 months, 16147 interviewees did not consume any alcohol (37.47% of total sample), 18346 had 1 alcohol consumption (42.57% of total sample), 8562 had 2 alcohol cxonsumption (19.87% of total sample), and 38 had 9 alcohol consumption (0.09% of total sample).

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

In [14]:

```
nesarc['S2AQ5B'] = pd.to_numeric(nesarc['S2AQ5B'], errors='coerce')
nesarc['S2AQ5B'].head(10)
```

Out[14]:

```
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
```

The first 10 rows of beer drinking frequency.

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

In [16]:

```
nesarc['S2AQ5B'] = nesarc['S2AQ5B'].astype('category') #set the data type as categorica
l data

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

p_beer_feq = nesarc['S2AQ5B'].value_counts(sort=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
Name: S2AQ5B, dtype: int64
```

In last 12 months, 836 interviewees drank 1 beer (4.56% of total sample), 645 had 2 beers (3.52% of total sample), 1535, 2190, 2451, 2603, 2127, 1194, 2268, 2442 had 3 to 10 beers respectively, they took up 8.37%, 11.94%, 13.36%, 14.19%, 11.59%, 6.51%, 12.36%, 13.31% of total sample. Also, 55 people drank 99 beers and they are 0.30% of total sample size.

```
percentages for S2AQ5B - usual frequency when drinking beer
1.0     0.045569
2.0     0.035158
3.0     0.083669
4.0     0.119372
5.0     0.133599
6.0     0.141884
7.0     0.115938
8.0     0.065082
9.0     0.123624
10.0    0.133108
99.0    0.002998
Name: S2AQ5B, dtype: float64
```

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

In [17]:

```
nesarc['S2AQ5D'] = pd.to_numeric(nesarc['S2AQ5D'], errors='coerce')
nesarc['S2AQ5D'] = nesarc['S2AQ5D'].astype("category")#check code - M
```

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

In [18]:

```
c_beer_quan = nesarc['S2AQ5D'].value_counts(sort=False)
print ('counts for S2AQ5D usual quantity when drink beer')
print(c_beer_quan)

p_beer_quan = nesarc['S2AQ5D'].value_counts(sort=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
Name: S2AQ5D, dtype: int64
percentages for S2AQ5D usual quantity when drink beer
1.0     0.388205
2.0     0.269159
3.0     0.139758
4.0     0.066718
5.0     0.027635
6.0     0.061485
7.0     0.006432
8.0     0.011174
9.0     0.001526
10.0    0.005887
11.0    0.000327
12.0    0.012591
13.0    0.000164
14.0    0.000327
15.0    0.001145
16.0    0.000055
17.0    0.000218
18.0    0.000981
20.0    0.000382
24.0    0.001254
25.0    0.000055
30.0    0.000164
36.0    0.000055
42.0    0.000055
99.0    0.004252
Name: S2AQ5D, dtype: float64
```

In the last 12 months, majority interviewees drank 6 or less beers on the day when they consume beer. And only about 3% of interviewees drank 7 or more beers. People who only drank 1 beer takes up the most percentage with 38.82% of total sample, followed by people drank 2 beers (26.92%) and 3 beers (13.98%).

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

In [19]:

```
#nesarc['TAB12MDX'] = pd.to_numeric(nesarc['TAB12MDX']) #convert variable to numeric
#nesarc['TAB12MDX'].head(25) #print the first 25 rows

#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
```

With ‘groupby’ method, get count of abuses due to alcohol in last 12 months. 1326 with one abuse, 25309 with two abuses, and 311 with 9 abuses.

In [20]:

```
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
```

Due to alcohol effects, in the last 12 months, 58.73% people experienced two abuses, 3.08% people had one abuse, and 0.72% had nine abuses.

P.s. It is different from the previous one because this method is taking NaN into account.

Obtain a subset of nesarc data, with the following criteria

Age from 26 to 50

Beer drinking status - S2AQ5A = Y

In [21]:

```
nesarc['AGE'] = pd.to_numeric(nesarc['AGE'])

#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
```

Interviewees’ ages range from 32 to 47 years old, they spread evenly as all age group takes around 3% to 4.8% of total sample size. 38 year olds make up the biggest age group with 504 people and 26 year olds with fewest people (325 interviewees).