5 rows × 3010 columns

# Importing libraries

In [1]:

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

# Read in the nesarc.csv file

In [2]:

*#read in csv file into*

nesarc = pd.read\_csv('nesarc.csv', low\_memory=**False**) *#increase efficiency*

# 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

# Printing the first 5 rows of nesarc

In [4]:

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

Out[4]:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Unnamed:**  **0** | **ETHRACE2A** | **ETOTLCA2** | **IDNUM** | **PSU** | **STRATUM** | **WEIGHT** | **C** |
| **0** | 0 | 5 |  | 1 | 4007 | 403 | 3928.613505 | 1 |
| **1** | 1 | 5 | 0.0014 | 2 | 6045 | 604 | 3638.691845 | 1 |
| **2** | 2 | 5 |  | 3 | 12042 | 1218 | 5779.032025 | 2 |
| **3** | 3 | 5 |  | 4 | 17099 | 1704 | 1071.754303 | 9 |
| **4** | 4 | 2 |  | 5 | 17099 | 1704 | 4986.952377 | 1 |

# 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

1. NaN
2. NaN
3. NaN

5 2.0

6 2.0

7 2.0

8 2.0

9 1.0

Name: S2BQ1B1, dtype: float64

# 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 |

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

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

1. NaN
2. NaN
3. NaN

5 2.0

6 2.0

7 2.0

8 1.0

9 2.0

Name: S2AQ5A, dtype: float64

# 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 |

# Convert HOW OFTEN DRANK BEER IN LAST 12

**MONTHS (S2AQ5B) to numeric & print first 10 rows**

nesarc['S2AQ5B'] = pd.to\_numeric(nesarc['S2AQ5B'], errors='coerce') nesarc['S2AQ5B'].head(10)

Out[14]:

0 NaN

1 10.0

1. NaN
2. NaN
3. NaN
4. NaN
5. NaN
6. NaN

8 9.0

9 NaN

Name: S2AQ5B, dtype: float64

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

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)

beer

|  |  |  |
| --- | --- | --- |
| 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 |  |
| percentages for S2AQ5B - usual frequency when drinking | | |
| 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 |

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

*#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 |

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 |

# 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 |