DATA MANAGEMENT

DATA MANAGEMENT

- Missing values
- Recoding
- Lab5
- Secondary variables
- Grouping variables
- lab 6

DATA MANAGEMENT

- One of the later steps in data management is evaluating whether you might want to create secondary variables.
- Secondary variables may include information from two or more primary variables.
- They can be created using a mathematical or logical operation on two or more variables.
- For our example we want to know the number of cigarettes smoked per month.
- We know how often each respondent smoked S3AQ3B1

DATA MANAGEMENT

- We can create a new variable USFREQMO which will give us an estimate of the quantity smoked per month.
- Again we use the variable S3AQ3B1 to estimate the number of cigarettes smoked in a month.
- Value 1: Everyday could estimate to 30 cigarettes per month
- Value 2: 5-6 days per week could estimate to 22 cigarettes per month
- Etc. for values 3, 4, 5, and 6

DATA MANAGEMENT

- These are estimates but they still capture the quantitative nature of the measure and also keep individuals ordered in terms of the frequency with which they smoke.
- The new variable is called USFREQMO

```
Recode1 = {1: 30, 2: 22, 3: 14, 4: 5, 5: 2.5, 6:1}
Sub2['USFREQMO'] = sub2['S3AQ3B1'].map(recode2)
```

DATA MANAGEMENT

- While it is a categorical variable we actually get more information out of it than we had originally been given.
- We can then request the frequency table for the sample of 1,706 young adults (>=18 and <=25) who smoked in the past year. This shows us more clearly the numbers and percentages of approximate number of days per month.

DATA MANAGEMENT

```
counts for USFREQMO
 30.0    1320
 14.0      91
 5.0       88
 1.0       71
 22.0      68
 2.5       65
NaN         3
Name: USFREQMO, dtype: int64
percentages for USFREQMO
30.0    0.775103
14.0    0.053435
5.0     0.051674
1.0     0.041691
22.0    0.039930
2.5     0.038168
Name: USFREQMO, dtype: float64
```

```
print('counts for USFREQMO')
p7=sub2[['USFREQMO'].value_counts(sort=False,dropna=False)
print(p7)
print('percentages for USFREQMO')
p8=sub2['USFREQMO'].value_counts(sort=False, normalize=True)
print(p8)
```

DATA MANAGEMENT

- We now know the estimated number of days smoked in a month in our new variable USFREQMO
- We also have the variable S3AQ3C1 which holds the usual quantity when smoked. 1-98 cigarettes, 99 unknown and blank.
- If we want to estimate the number of cigarettes that participants smoked per month it would make sense to multiply the two variables and get a product that represents the number of cigarettes smoked per month.
- The new variable will be called NUMCIGMO_EST

DATA MANAGEMENT

```
sub2['NUMCIGMO_EST'] = sub2['USFREQMO'] * sub2['S3AQ3C1']

sub3= sub2[['IDNUM','S3AQ3C1','USFREQMO','NUMCIGMO_EST']]
sub3.head(25)
```

What does this do?

DATA MANAGEMENT

```
In [8]: subset3.head(25)
Out[8]:
     IDNUM  S3AQ3C1  USFREQMO  NUMCIGMO_EST
20      21        3      30.0          90.0
76      77        3      22.0          66.0
102    103       10      30.0         300.0
121    122       10      30.0         300.0
135    136       20      30.0         600.0
149    150        5      30.0         150.0
154    155        8      30.0         240.0
173    174        1      30.0          30.0
177    178       10      30.0         300.0
183    184       20      30.0         600.0
187    188        2       5.0          10.0
209    210        3      30.0          90.0
219    220        5      14.0          70.0
222    223        1      30.0          30.0
278    279       98      30.0        2940.0
336    337       20      30.0         600.0
363    364       20      30.0         600.0
398    399        2      22.0          44.0
412    413        5      30.0         150.0
417    418       20      30.0         600.0
508    509       30      30.0         900.0
511    512        1       2.5           2.5
519    520       20      30.0         600.0
522    523       10      30.0         300.0
529    530        4      30.0         120.0
```

DATA MANAGEMENT

- Once you have created secondary variables such as USFREQMO and NUMCIGMO_EST you can then consider whether any of your quantitative variables or categorical variables need to be further grouped or binned.
- Currently AGE is a quantitative variable in the NESARC dataset, if we want to compare age groups categorically we need to group the values into categories.
- This would allow us to compare the number and percent of observations at each age group.

DATA MANAGEMENT

```
#quartile split qcut function into 4 groups

print('AGE - 4 Categories - quartiles')

subset2['AGEGROUP'] = pandas.qcut(subset2.AGE, 4,
labels=['1=25%tile','2=50%tile','3=75%tile','4=100%tile'])

c14= subset2['AGEGROUP'].value_counts(sort=False, dropna=True)

print(c14)
```

DATA MANAGEMENT

```
AGE - 4 Categories - quartiles
1=25%tile     582
2=50%tile     467
3=75%tile     231
4=100%tile    426
Name: AGEGROUP, dtype: int64

In [2]:
```

DATA MANAGEMENT

- Custom splits can also be set if you use the cut() function.
- This allows you to choose what every grouping you wish to use.
- The function can take several parameters (ref to pandas docs)
- In this example we pass the integer values that set up the bins.
- The first integer is one less than the first value in the bin (17 instead of 18)

DATA MANAGEMENT

```
#categorise variable based on customised splits using the cut()
functions

# splits into three groups, 18-20, 21-22, and 23-25

subset2['AGEGROUP2']= pandas.cut(subset2.AGE, [17, 20, 22, 25],
labels=['18-20','21-22','23-25'])

c15 = subset2['AGEGROUP2'].value_counts(sort=False, dropna=True)

print(c15)
```

DATA MANAGEMENT

```
18-20    582
21-22    467
23-25    657
Name: AGEGROUP2, dtype: int64

In [5]:
```

DATA MANAGEMENT

- We can now simply compare the two variables AGE and AGEGROUP2 in a crosstab.

```
print(pandas.crosstab(subset2['AGEGROUP2'],subset2['AGE']))
```

```
In [9]: print(pandas.crosstab(subset2['AGEGROUP2'],subset2['AGE']))
AGE          18   19   20   21   22   23   24   25
AGEGROUP2
18-20       161  200  221    0    0    0    0    0
21-22         0    0    0  239  228    0    0    0
23-25         0    0    0    0    0  231  241  185
```

The count of observations within each age group can be seen. This also serves as a check to make sure your grouping worked as expected.