Practice notebook for univariate analysis using NHANES data

This notebook will give you the opportunity to perform some univariate analyses on your own using the NHANES. These analyses are similar to what was done in the week 2 NHANES case study notebook.

You can enter your code into the cells that say "enter your code here", and you can type responses to the questions into the cells that say "Type Markdown and Latex".

Note that most of the code that you will need to write below is very similar to code that appears in the case study notebook. You will need to edit code from that notebook in small ways to adapt it to the prompts below.

To get started, we will use the same module imports and read the data in the same way as we did in the case study:

In [15]:

```
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import statsmodels.api as sm
import numpy as np

da = pd.read_csv("nhanes_2015_2016.csv")
da.columns
```

Out[15]:

```
Index(['SEQN', 'ALQ101', 'ALQ110', 'ALQ130', 'SMQ020', 'RIAGENDR', 'RIDAGEYR',
       'RIDRETH1', 'DMDCITZN', 'DMDEDUC2', 'DMDMARTL', 'DMDHHSIZ', 'WTINT2YR',
       'SDMVPSU', 'SDMVSTRA', 'INDFMPIR', 'BPXSY1', 'BPXDI1', 'BPXSY2',
       'BPXDI2', 'BMXWT', 'BMXHT', 'BMXBMI', 'BMXLEG', 'BMXARML', 'BMXARMC',
       'BMXWAIST', 'HIQ210'],
      dtype='object')
```

Question 1

Relabel the marital status variable DMDMARTL (https://wwwn.cdc.gov/Nchs/Nhanes/2015-2016/DEMO_I.htm#DMDMARTL) to have brief but informative character labels. Then construct a frequency table of these values for all people, then for women only, and for men only. Then construct these three frequency tables using only people whose age is between 30 and 40

In [18]:

```
# insert your code here
print(da.DMDMARTL.value_counts(),'\n')
da["DMDMARTL"] = da.DMDMARTL.replace({1: "Married", 2: "Widowed", 3: "Divorced",
                                      4: "Separated", 5: "Never married", 6: "Living with partner",
                                      77: "Unknown", 99: "Unknown"})
da["DMDMARTL"] = da.DMDMARTL.fillna("Unknown")
freq_table = da.DMDMARTL.value_counts()
print(freq_table, '\n')
da["RIAGENDR"] = da.RIAGENDR.replace({1: "Male", 2: "Female"})
da["RIAGENDR"] = da.RIAGENDR.fillna("Unknown")
males = da[da["RIAGENDR"] == "Male"]
females = da[da["RIAGENDR"] == "Female"]
print(da.RIAGENDR.value_counts(),'\n')
print(da.groupby("RIAGENDR")["DMDMARTL"].value_counts(),'\n')
thirty_forty = pd.DataFrame(da[da["RIDAGEYR"] >= 30])
thirty_forty = thirty_forty[thirty_forty["RIDAGEYR"] <= 40]
#print(thirty_forty.head())
thirty_forty.groupby("RIAGENDR")["DMDMARTL"].value_counts()
```

```
1.0     2780
5.0     1004
3.0      579
6.0      527
2.0      396
4.0      186
77.0       2
Name: DMDMARTL, dtype: int64

Married                2780
Never married          1004
Divorced                579
Living with partner     527
Widowed                 396
Unknown                 263
Separated               186
Name: DMDMARTL, dtype: int64

Female    2976
Male      2759
Name: RIAGENDR, dtype: int64

RIAGENDR  DMDMARTL
Female    Married                1303
          Never married           520
          Divorced                350
          Widowed                 296
          Living with partner     262
          Unknown                 127
          Separated               118
Male      Married                1477
          Never married           484
          Living with partner     265
          Divorced                229
          Unknown                 136
          Widowed                 100
```

```
            Separated                68
Name: DMDMARTL, dtype: int64
```

```
Out[18]: RIAGENDR  DMDMARTL
         Female    Married                 285
                   Never married           116
                   Living with partner      65
                   Divorced                 46
                   Separated                18
                   Widowed                   2
         Male      Married                 275
                   Never married           101
                   Living with partner      78
                   Divorced                 24
                   Separated                12
                   Widowed                   3
                   Unknown                   1
         Name: DMDMARTL, dtype: int64
```

Q1a. Briefly comment on some of the differences that you observe between the distribution of marital status between women and men, for people of all ages.

Fewer women married, divorced; More women never married

Q1b. Briefly comment on the differences that you observe between the distribution of marital status states for women between the overall population, and for women between the ages of 30 and 40.

More women married, divorced;

Q1c. Repeat part b for the men.

Question 2

Restricting to the female population, stratify the subjects into age bands no wider than ten years, and construct the distribution of marital status within each age band. Within each age band, present the distribution in terms of proportions that must sum to 1.

In [69]:
```
females["AGEGROUP"] = pd.cut(females.RIDAGEYR, [10, 20, 30, 40, 50, 60, 70, 80])
dx = females.groupby(["AGEGROUP"])["RIAGENDR"]
print(dx.value_counts(), '\n')
dx.value_counts() / (females.shape [1] * 100)
```

```
AGEGROUP  RIAGENDR
(10, 20]  Female      165
(20, 30]  Female      514
(30, 40]  Female      474
(40, 50]  Female      502
(50, 60]  Female      470
(60, 70]  Female      441
(70, 80]  Female      410
Name: RIAGENDR, dtype: int64
```

Out[69]:
```
AGEGROUP  RIAGENDR
(10, 20]  Female      0.056897
(20, 30]  Female      0.177241
(30, 40]  Female      0.163448
(40, 50]  Female      0.173103
(50, 60]  Female      0.162069
(60, 70]  Female      0.152069
(70, 80]  Female      0.141379
Name: RIAGENDR, dtype: float64
```

Q2a. Comment on the trends that you see in this series of marginal distributions.

Other than 10-20, pretty evenly distributed

Q2b. Repeat the construction for males.

In [70]:
```
# insert your code here
males["AGEGROUP"] = pd.cut(males.RIDAGEYR, [10, 20, 30, 40, 50, 60, 70, 80])
dx = males.groupby(["AGEGROUP"])["RIAGENDR"]
print(dx.value_counts(), '\n')
dx.value_counts() / (females.shape [1] * 100)
```

```
AGEGROUP  RIAGENDR
(10, 20]  Male        175
(20, 30]  Male        432
(30, 40]  Male        458
(40, 50]  Male        401
(50, 60]  Male        454
(60, 70]  Male        437
(70, 80]  Male        402
Name: RIAGENDR, dtype: int64
```

Out[70]:
```
AGEGROUP  RIAGENDR
(10, 20]  Male        0.060345
(20, 30]  Male        0.148966
(30, 40]  Male        0.157931
(40, 50]  Male        0.138276
(50, 60]  Male        0.156552
(60, 70]  Male        0.150690
(70, 80]  Male        0.138621
Name: RIAGENDR, dtype: float64
```

Q2c. Comment on any notable differences that you see when comparing these results for females and for males.

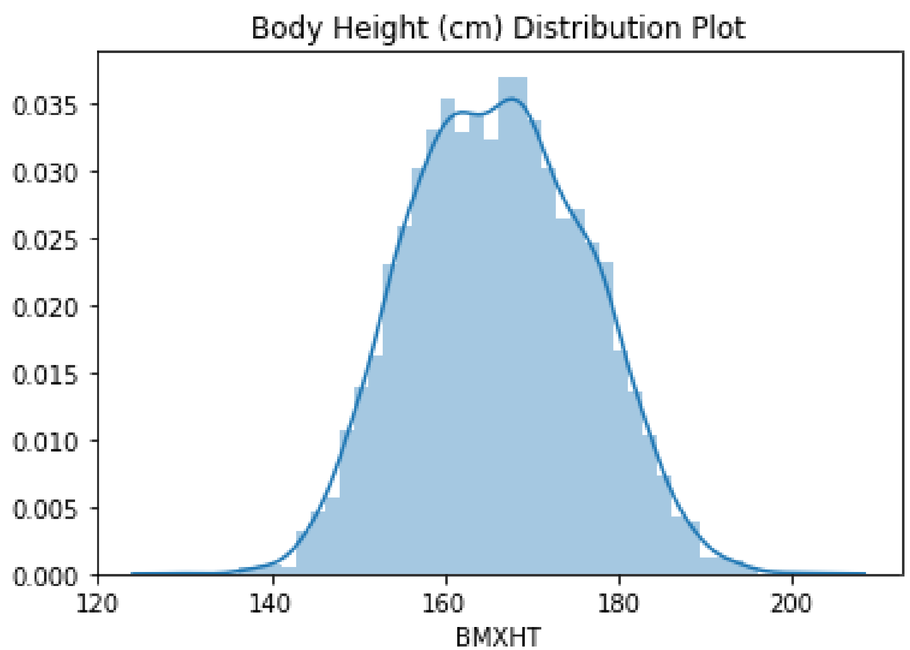
ditto

Question 3

Construct a histogram of the distribution of heights using the BMXHT variable in the NHANES sample.

In [71]:
```
# insert your code here
sns.distplot(da.BMXHT.dropna()).set_title('Body Height (cm) Distribution Plot')
```

Out[71]:
```
Text(0.5,1,'Body Height (cm) Distribution Plot')
```

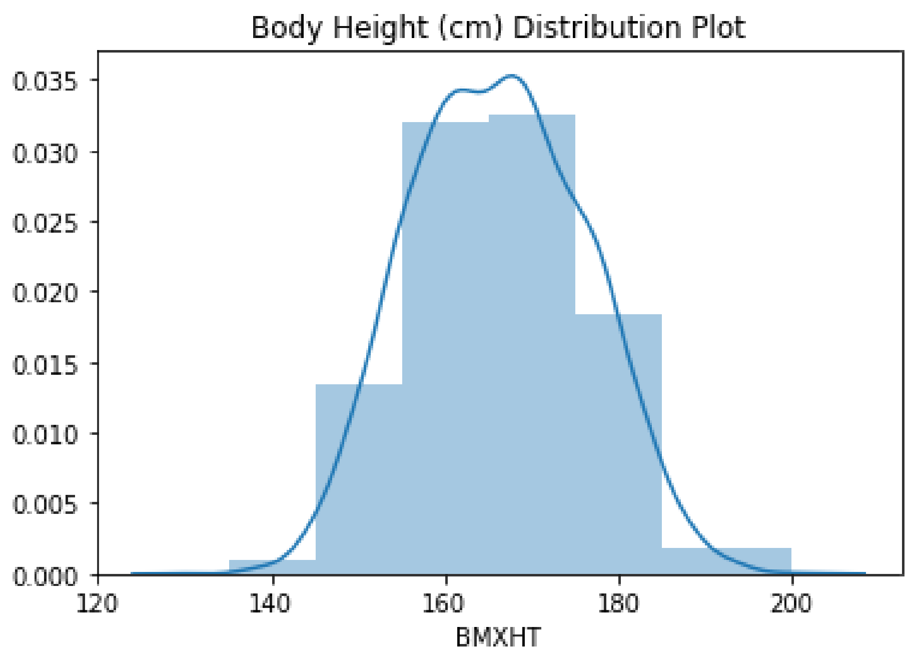


Q3a. Use the `bins` argument to [distplot (https://seaborn.pydata.org/generated/seaborn.distplot.html)](https://seaborn.pydata.org/generated/seaborn.distplot.html) to produce histograms with different numbers of bins. Assess whether the default value for this argument gives a meaningful result, and comment on what happens as the number of bins grows excessively large or excessively small.

In [73]:
```
da["HEIGHTGRP"] = pd.cut(da.BMXHT, [135, 145, 155, 165, 175, 185, 200]) # Create age strata based on these cut points
print("hello")
print(da.HEIGHTGRP.value_counts(), '\n')
sns.distplot(da.BMXHT.dropna(), bins=[135, 145, 155, 165, 175, 185, 200]).set_title('Body Height (cm) Distribution Plot')
```

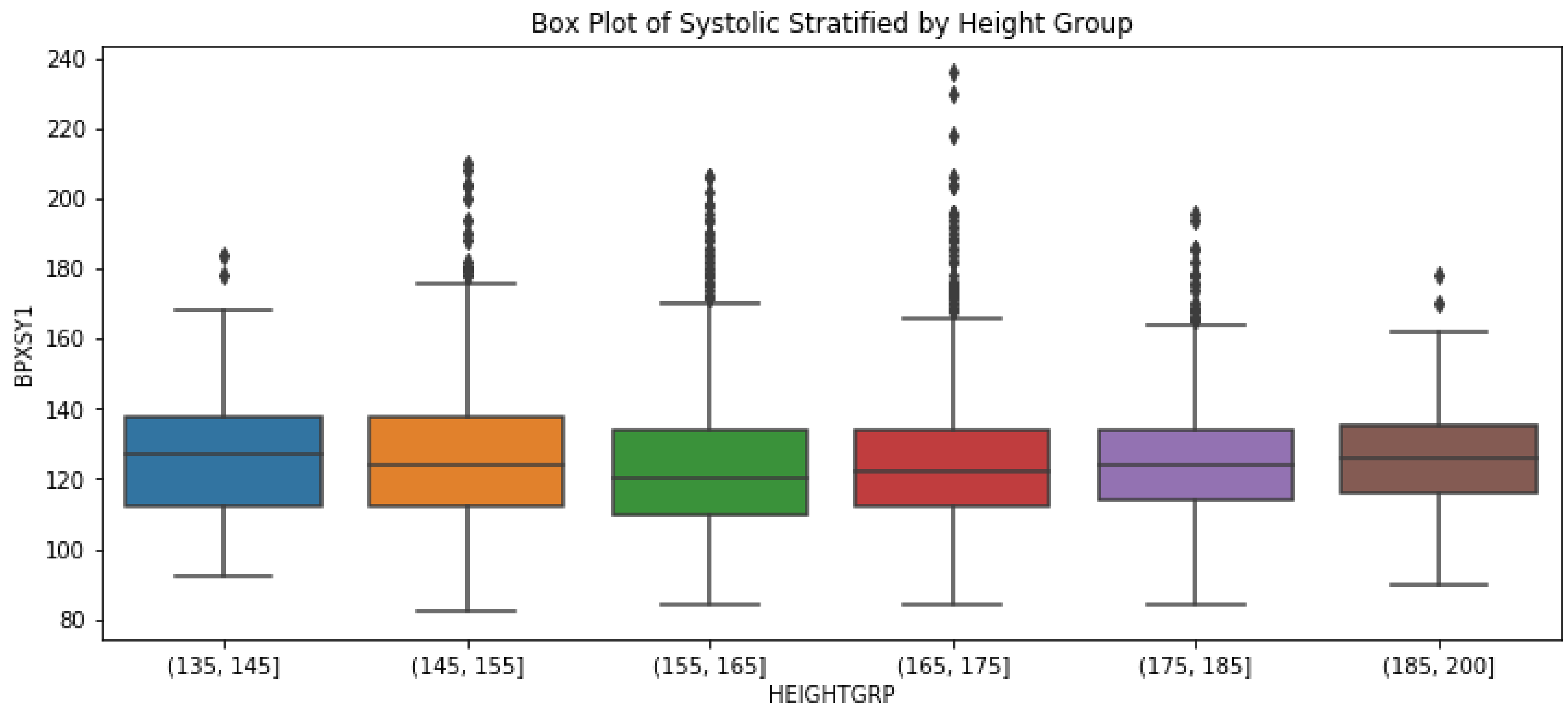
```
hello
(165, 175]    1850
(155, 165]    1816
(175, 185]    1026
(145, 155]     772
(185, 200]     148
(135, 145]      58
Name: HEIGHTGRP, dtype: int64
```

Out[73]:
```
Text(0.5,1,'Body Height (cm) Distribution Plot')
```



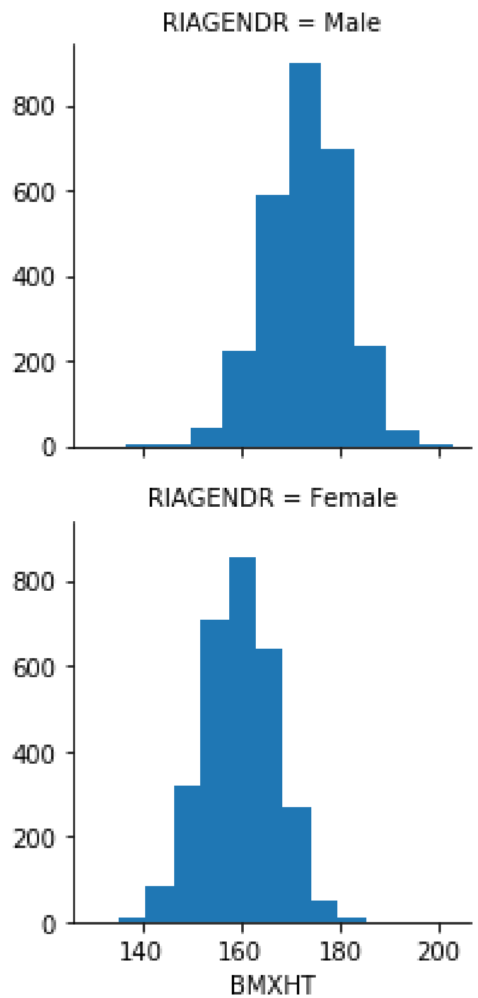
In [74]:
```
plt.figure(figsize=(12, 5))  # Make the figure wider than default (12cm wide by 5cm tall)
sns.boxplot(x="HEIGHTGRP", y="BPXSY1", data=da).set_title('Box Plot of Systolic Stratified by Height Group')
```

Out[74]: Text(0.5,1,'Box Plot of Systolic Stratified by Height Group')



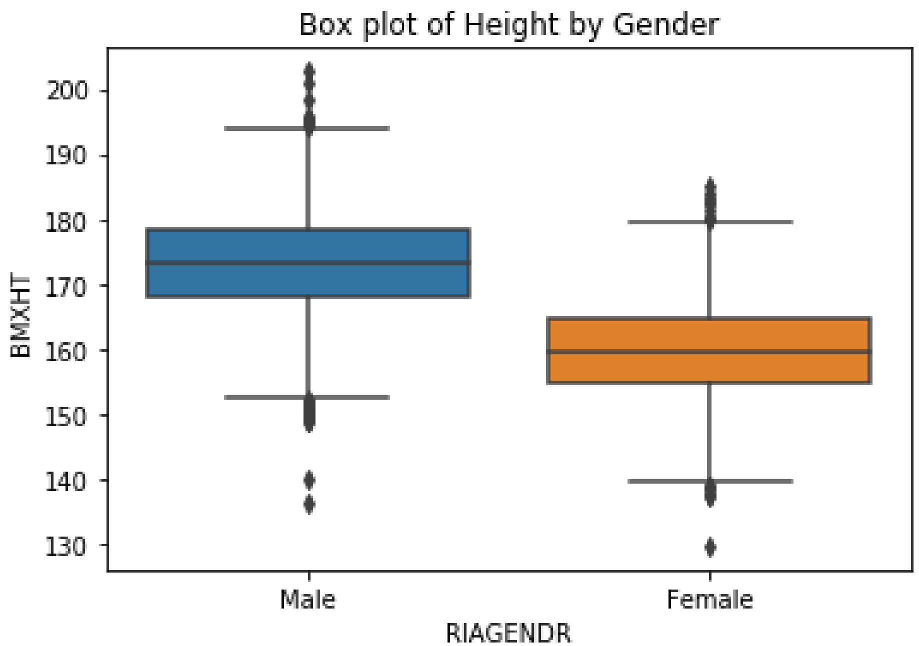
Q3b. Make separate histograms for the heights of women and men, then make a side-by-side boxplot showing the heights of women and men.

In [75]:
```
# insert your code here
g = sns.FacetGrid(da, row = "RIAGENDR")
g = g.map(plt.hist, "BMXHT")
plt.show()
```



In [76]:
```
sns.boxplot(x = da["RIAGENDR"], y = da["BMXHT"]).set_title("Box plot of Height by Gender")
```

Out[76]:
```
Text(0.5,1,'Box plot of Height by Gender')
```



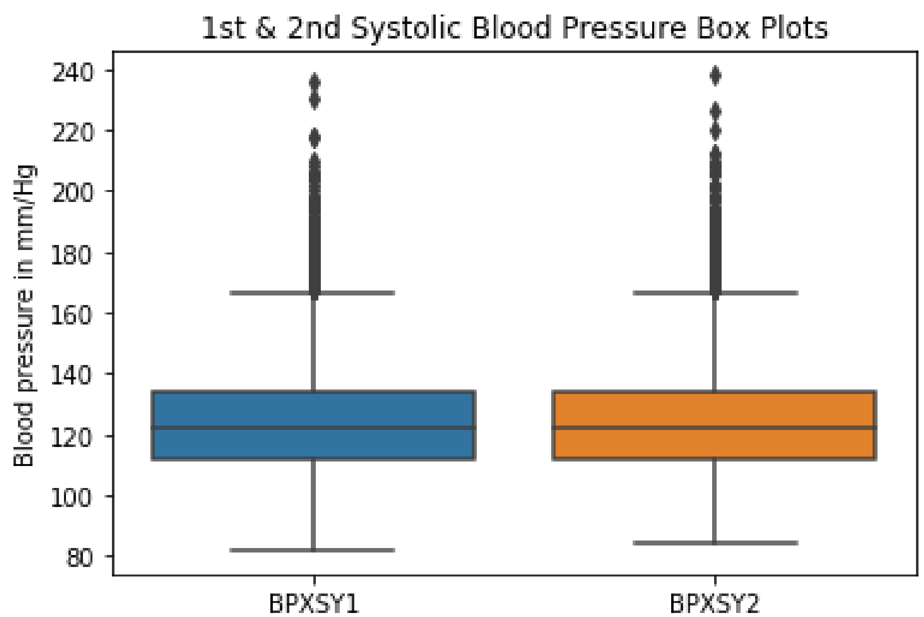
Q3c. Comment on what features, if any are not represented clearly in the boxplots, and what features, if any, are easier to see in the boxplots than in the histograms.

It's a bit easier to tell the males are generally taller than females. Scientific breakthrough!!

Question 4

Make a boxplot showing the distribution of within-subject differences between the first and second systolic blood pressure measurents (BPXSY1 (https://wwwn.cdc.gov/Nchs/Nhanes/2015-2016/BPX_I.htm#BPXSY1) and BPXSY2 (https://wwwn.cdc.gov/Nchs/Nhanes/2015-2016/BPX_I.htm#BPXSY2)).

In [79]:
```
# insert your code here
# kindly define "within-subject differences"
bp = sns.boxplot(data=da.loc[:, ["BPXSY1", "BPXSY2"]])
bp.set_ylabel("Blood pressure in mm/Hg")
bp.set_title('1st & 2nd Systolic Blood Pressure Box Plots')
plt.show()
```



Q4a. What proportion of the subjects have a lower SBP on the second reading compared to the first?

In [ ]:
```
# insert your code here
# very difficult to tell
```

Q4b. Make side-by-side boxplots of the two systolic blood pressure variables.

In [4]:

```
# insert your code here
# What I did in step 1
```

Q4c. Comment on the variation within either the first or second systolic blood pressure measurements, and the variation in the within-subject differences between the first and second systolic blood pressure measurements.

Learn how to define what you want better! Very imprecise!

Question 5

Construct a frequency table of household sizes for people within each educational attainment category (the relevant variable is [DMDEDUC2 (https://wwwn.cdc.gov/Nchs/Nhanes/2015-2016/DEMO_I.htm#DMDEDUC2)](https://wwwn.cdc.gov/Nchs/Nhanes/2015-2016/DEMO_I.htm#DMDEDUC2)). Convert the frequencies to proportions.

In [25]:
```
# insert your code here
educ_freq_table = da.copy()
educ_freq_table["AGEGROUP"] = pd.cut(educ_freq_table.RIDAGEYR, [10, 20, 30, 40, 50, 60, 70, 80])
print(da.DMDEDUC2.value_counts(),'\n')
educ_freq_table["DMDEDUC2"] = educ_freq_table.DMDEDUC2.replace({1: "<9", 2: "9-11", 3: "HS/GED",
                                                              4: "Some college/AA", 5: "College",
                                                              7: "Refused", 9: "Unknown"})
print(educ_freq_table.DMDEDUC2.value_counts() / da.DMDEDUC2.value_counts().sum(), '\n')
```

```
4.0    1621
5.0    1366
3.0    1186
1.0     655
2.0     643
9.0       3
Name: DMDEDUC2, dtype: int64

Some college/AA    0.296127
College            0.249543
HS/GED             0.216661
<9                 0.119657
9-11               0.117464
Unknown            0.000548
Name: DMDEDUC2, dtype: float64
```

Q5a. Comment on any major differences among the distributions.

25% of the population has completed college; 75% has at least a high school education

Q5b. Restrict the sample to people between 30 and 40 years of age. Then calculate the median household size for women and men within each level of educational attainment.

In [26]:
```
# insert your code here
educ_freq_table.loc[educ_freq_table.AGEGROUP == pd.Interval(left=30, right=40)].groupby(['RIAGENDR','DMDEDUC2'
]).DMDHHSIZ.median()
```

Out[26]:
```
RIAGENDR  DMDEDUC2
Female    9-11               5
          <9                 5
          College            4
          HS/GED             5
          Some college/AA    4
Male      9-11               5
          <9                 5
          College            3
          HS/GED             4
          Some college/AA    4
Name: DMDHHSIZ, dtype: int64
```

Question 6

The participants can be clustered into "maked variance units" (MVU) based on every combination of the variables SDMVSTRA (https://wwwn.cdc.gov/Nchs/Nhanes/2015-2016/DEMO_I.htm#SDMVSTRA) and SDMVPSU (https://wwwn.cdc.gov/Nchs/Nhanes/2015-2016/DEMO_I.htm#SDMVPSU). Calculate the mean age (RIDAGEYR (https://wwwn.cdc.gov/Nchs/Nhanes/2015-2016/DEMO_I.htm#RIDAGEYR)), height (BMXHT (https://wwwn.cdc.gov/Nchs/Nhanes/2015-2016/BMX_I.htm#BMXHT)), and BMI (BMXBMI (https://wwwn.cdc.gov/Nchs/Nhanes/2015-2016/BMX_I.htm#BMXBMI)) for each gender (RIAGENDR (https://wwwn.cdc.gov/Nchs/Nhanes/2015-2016/DEMO_I.htm#RIAGENDR)), within each MVU, and report the ratio between the largest and smallest mean (e.g. for height) across the MVUs.

In [1]:
```
# insert your code here
# yeah, sure
```

Q6a. Comment on the extent to which mean age, height, and BMI vary among the MVUs.

Q6b. Calculate the inter-quartile range (IQR) for age, height, and BMI for each gender and each MVU. Report the ratio between the largest and smalles IQR across the MVUs.

In [ ]:
```
# insert your code here
```

Q6c. Comment on the extent to which the IQR for age, height, and BMI vary among the MVUs.