

a1-python

● Graded

Student

Tagore Kosireddy

Total Points

73.25 / 75 pts

Autograder Score

13.0 / 13.0

Passed Tests

Public Tests

q0 (2/2)

q1a (5/5)

q1c (6/6)

Question 2

Q1a-Manual

2 / 2 pts

✓ - 0 pts Correct

Question 3

Q1b

12 / 12 pts

✓ - 0 pts Correct

✓ - 0 pts ST - detailed

Question 4

Q2a

9.25 / 10 pts

✓ - 0.75 pts Plots do not fit in view on Gradescope.
(width longer than the length of cell)

Question 5

Q2b

10 / 10 pts

✓ - 0 pts Correct

Question 6

Q2c

10 / 10 pts

✓ - 0 pts Correct

Question 7

Q2d

9 / 10 pts

✓ - 1 pt Updated

Data and labels do not match

✓ - 0 pts plot does not match requested results; different norms

Question 8

General

8 / 8 pts

✓ - 0 pts To Everyone:

When submitting on Gradescope, make sure to check:

1. Your notebook runs in the Autograder
2. Your notebooks looks correct in its formatting.

✓ - 0 pts Code is not commented

Add good comments to your code.

Autograder Results

Autograder Output

```

      _____
 /  _____ \  _ |  |  _ |  |  |  |
| /    \  |  |  |  |  |  |  |  |  |  | |
| |      | |  |  |  |  |  |  |  |  |
| |      | |  |  |  |  |  |  |  |  |
| |      | |  |  |  |  |  |  |  |  |
| \    /  |  |  |  |  |  |  |  |  |
 \  _____ /   \  |  |  |  |  |  |
   \  _____ /   \  |  |  |  |  |
                        \  |  |  |  |  |
                          \  |  |  |  |
                            \  |  |  |
                              \  |  |
                                \  |
                                  \
v5.2.3

```

----- GRADING SUMMARY -----

Score for q1c (3.000) differs from logged score (6.000)

Total Score: 13.000 / 13.000 (100.000%)

	name	score	max_score
0	Public Tests	NaN	NaN
1	q0	2.0	2.0
2	q1a	5.0	5.0
3	q1c	6.0	6.0

Public Tests

q0 results: All test cases passed!

q1a results: All test cases passed!

q1c results: All test cases passed!

q1a results: All test cases passed!

q1c results: All test cases passed!

q0 (2/2)

q0 results: All test cases passed!

q1a (5/5)

q1a results: All test cases passed!

q1c (6/6)

q1c results: All test cases passed!

Submitted Files

a1 - Python

This assignment will cover some questions related to topics of data types, attribute types, and exploratory data analysis. The assignment will also serve as a further introduction to using a high-level language for analysis (e.g., R, Python).

Make sure that you keep this notebook named as "a1.ipynb"

Submit the zip-file created after running your notebook on the Linux lab machines.

Your answers must be computer generated (including text and diagrams). Your final document submission should include text responses to questions and description of your efforts, tables, R/Python code used to calculate answers, and figures.

Any other packages or tools, outside those listed in the assignments or Canvas, should be cleared by Dr. Brown before use in your submission.

Q0 - Setup

The following code looks to see whether your notebook is run on Gradescope (GS), Colab (COLAB), or the linux Python environment you were asked to setup.

In [36]:

```
import re
import os
import platform
import sys

# flag if notebook is running on Gradescope
if re.search(r'amzn', platform.uname().release):
    GS = True
else:
    GS = False

# flag if notebook is running on Colaboratory
try:
    import google.colab
    COLAB = True
```

```

except:
    COLAB = False

# flag if running on Linux lab machines.
cname = platform.uname().node
if re.search(r'(guardian|colossus|c28)', cname):
    LLM = True
else:
    LLM = False

print("System: GS - %s, COLAB - %s, LLM - %s" % (GS, COLAB, LLM))

```

System: GS - False, COLAB - False, LLM - True

Notebook Setup

It is good practice to list all imports needed at the top of the notebook. You can import modules in later cells as needed, but listing them at the top clearly shows all which are needed to be available / installed.

If you are doing development on Colab, the otter-grader package is not available, so you will need to install it with pip (uncomment the cell directly below).

In [37]:

```

# Only uncomment if you developing on Colab
# if COLAB == True:
#     print("Installing otter:")
#     !pip install otter-grader==4.2.0

```

In [38]:

```

# Import standard DS packages
import pandas as pd
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

# Package for Autograder
import otter
grader = otter.Notebook()

```

In [39]: `grader.check("q0")`

Out [39]: q0 results: All test cases passed!

Q1 - Census Data

The ACSIncome dataset is one of five datasets created by Ding et al. [1](#) as an improved alternative to the popular UCI Adult dataset [2](#).

You can explore the files a bit in a text editor to understand the format. Note, there are about 1.6M rows of data.

Use the column labels as described on the [reference website](#), e.g., 'AGEP', 'COW', etc.

References

[1](#): Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. Retiring adult: new datasets for fair machine learning. Advances in neural information processing systems, 34:6478–6490, 2021.

[2](#): Ronny Kohavi and Barry Becker. Adult data set. UCI Machine Learning Repository, 1996.

DOI: <https://doi.org/10.24432/C5XW20>. URL: <https://archive.ics.uci.edu/ml/datasets/adult>.

Q1a - Load Data

Load in the data from the csv-files into a Pandas Data Frame. Make sure to recognize any missing values when the data is read in.

Use the column labels as described on the [reference website](#), e.g., 'AGEP', 'COW', etc.

```
In [40]: # Reading data
income = pd.read_csv("ACSIncome.csv", names = ['AGEP', 'COW', 'SCHL', 'MAR',
'OCCP', 'POBP', 'RELP', 'WKHP', 'SEX', 'RAC1P', 'ST', 'PINC'])
if income[pd.isnull(income).any(axis = 1)].shape[0] == 0:
    print("There is no any missing values")
else:
    print("There is a missing value")

income.head()
```

There is no any missing values

```
Out [40]:      AGEP COW SCHL MAR  OCCP POBP RELP WKHP SEX RAC1P ST  PINCP
0  18.0  1.0  18.0  5.0 4720.0 13.0 17.0 21.0 2.0   2.0  1.0 1600.0
1  53.0  5.0  17.0  5.0 3605.0 18.0 16.0 40.0 1.0   1.0  1.0 10000.0
2  41.0  1.0  16.0  5.0 7330.0  1.0 17.0 40.0 1.0   1.0  1.0 24000.0
3  18.0  6.0  18.0  5.0 2722.0  1.0 17.0  2.0 2.0   1.0  1.0  180.0
4  21.0  5.0  19.0  5.0 3870.0 12.0 17.0 50.0 1.0   1.0  1.0 29000.0
```

```
In [41]: grader.check("q1a")
```

Out [41]: q1a results: All test cases passed!

Q1b - Variable Definitions

To answer this question, you may have to do a bit of reading and research into this data set. If you can not find a clear explanation of what a variable is and how it is defined say so.

Describe what each row represents in the data set.

For each variable (column of the data set), write a brief, clear 1-sentence description (less than one line long) of what the variable is, i.e., what information does it describe and how is it defined or collected.

For example, the variable `AGEP` could be described as:

- **AGEP** is the age of an individual; the value is reported in integer units of years, 0-99.

Refer to each variables by the column name.

YOUR ANSWERS

A row represents ...

- **AGEP** is the age of an individual; the value is reported in integer units of years.
- **COW** is the class of worker; the value is reported in intergers(0-9) each represents different class of workers.
- **SCHL** is the Educational attainment; value is reported in intergers(1-24) each represents education they completed.
- **MAR** is the Marital status; the value is reported in intergers(1-5) each representing the individual martial status.

- **OCCP** is the Occupation; the value is reported in integer units for each occupation code.
- **POBP** is the Place of birth; the value is reported in integer number for each states.
- **RELP** is the Relationship to householder; the value is reported in integers(1-17) each represents different relationship.
- **WKHP** is the Usual hours worked per week in the past 12 months; Values are an integer from 1 to 99, Any hours above 99 are rounded down to 99.
- **SEX** is the Sex code; the value is reported in integers(1,2) 1 representing male and 2 representing female.
- **RAC1P** is the Race code; the value is reported in integers(1-9) each representing different race.
- **ST** is the state code, the value is reported in integers(1-72) each representing different state.
- **PINCP** is the Total annual income per person; the value is reported in integers ranging from 104-1,423,000.

Q1c - Attribute Types

To answer this question, you may again have to do a bit of reading and research into this data set. If you can not find a clear explanation of what a variable is and how it is defined say so.

For each variable, state the attribute type: 1- *nominal*, 2- *ordinal*, 3- *interval*, or 4- *ratio*.

For example, the variable `AGEP` could be described as:

- **AGEP** is the age of an individual; the value is reported in integer units of years, 0-99.
Ratio

Therefore, in the code cell below you would have:

```
type_agep = 4
```

In [42]:

```
type_agep = 4
type_cow = 1
type_schl = 2
type_mar = 1
type_occp = 1
type_pobp = 1
type_relp = 1
type_wkhp = 4
type_sex = 1
type_rac1p = 1
```

```
type_st = 1
type_pincp = 4
```

In [43]: `grader.check("q1c")`

Out [43]: q1c results: All test cases passed!

Q2 - Exploratory Data Analysis

We will explore aspects of the income data from above.

REMINDER! For each part of the question below, make sure the figure is of reasonable size. You may want to consider using the `figsize` parameter in matplotlib.

Use good visualization practices as discussed in class and in the [Visualization book reference](#). Make sure to label everything.

Q2a - Visualize: Amounts: Single Variable

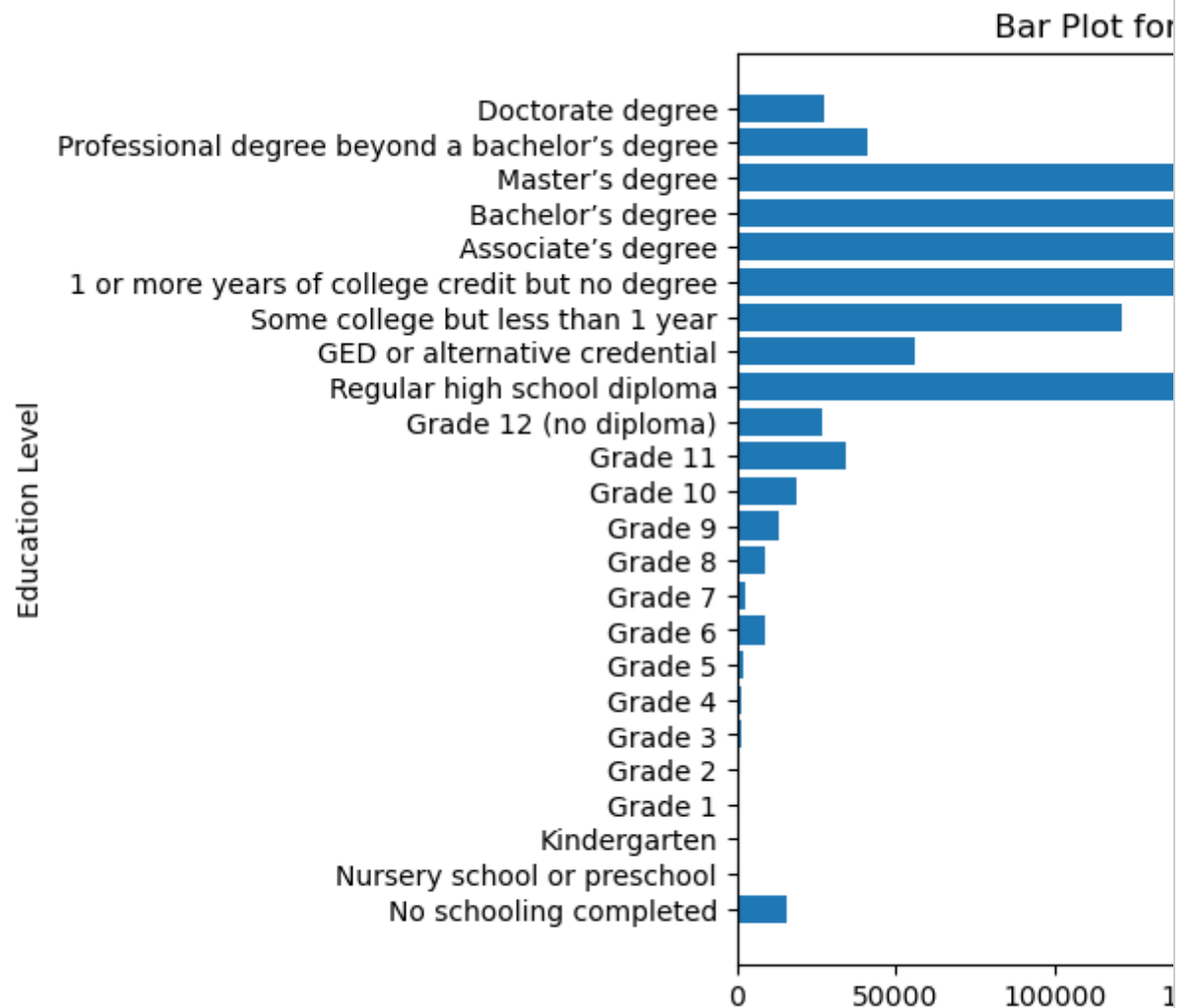
Create a bar plot of `SCHL`.

In [44]:

```
# Create bar plot of education
# Source : https://www.geeksforgeeks.org/matplotlib-pyplot-barh-function-in-python/
plt.figure(figsize = (8,6))
# Find frequency of SCHL for various educational levels
frequency_of_SCHL = income.SCHL.value_counts().sort_index()
# labels for SCHL
labels_for_SCHL = ["No schooling completed", "Nursery school or preschool", "Kindergarten",
                  "Grade 1", "Grade 2", "Grade 3", "Grade 4", "Grade 5",
                  "Grade 6", "Grade 7", "Grade 8", "Grade 9", "Grade 10",
                  "Grade 11", "Grade 12 (no diploma)", "Regular high school diploma",
                  "GED or alternative credential", "Some college but less than 1 year",
                  "1 or more years of college credit but no degree",
                  "Associate's degree", "Bachelor's degree", "Master's degree",
                  "Professional degree beyond a bachelor's degree", "Doctorate degree"]

# Bar plot for Education of an Individual
plt.barh(y = frequency_of_SCHL.index, width = frequency_of_SCHL)
plt.xlabel("Count")
plt.ylabel("Education Level")
```

```
plt.title(" Bar Plot for Education Level of an Individual")
plt.xticks(frequency_of_SCHL.index, labels_for_SCHL)
plt.show()
```



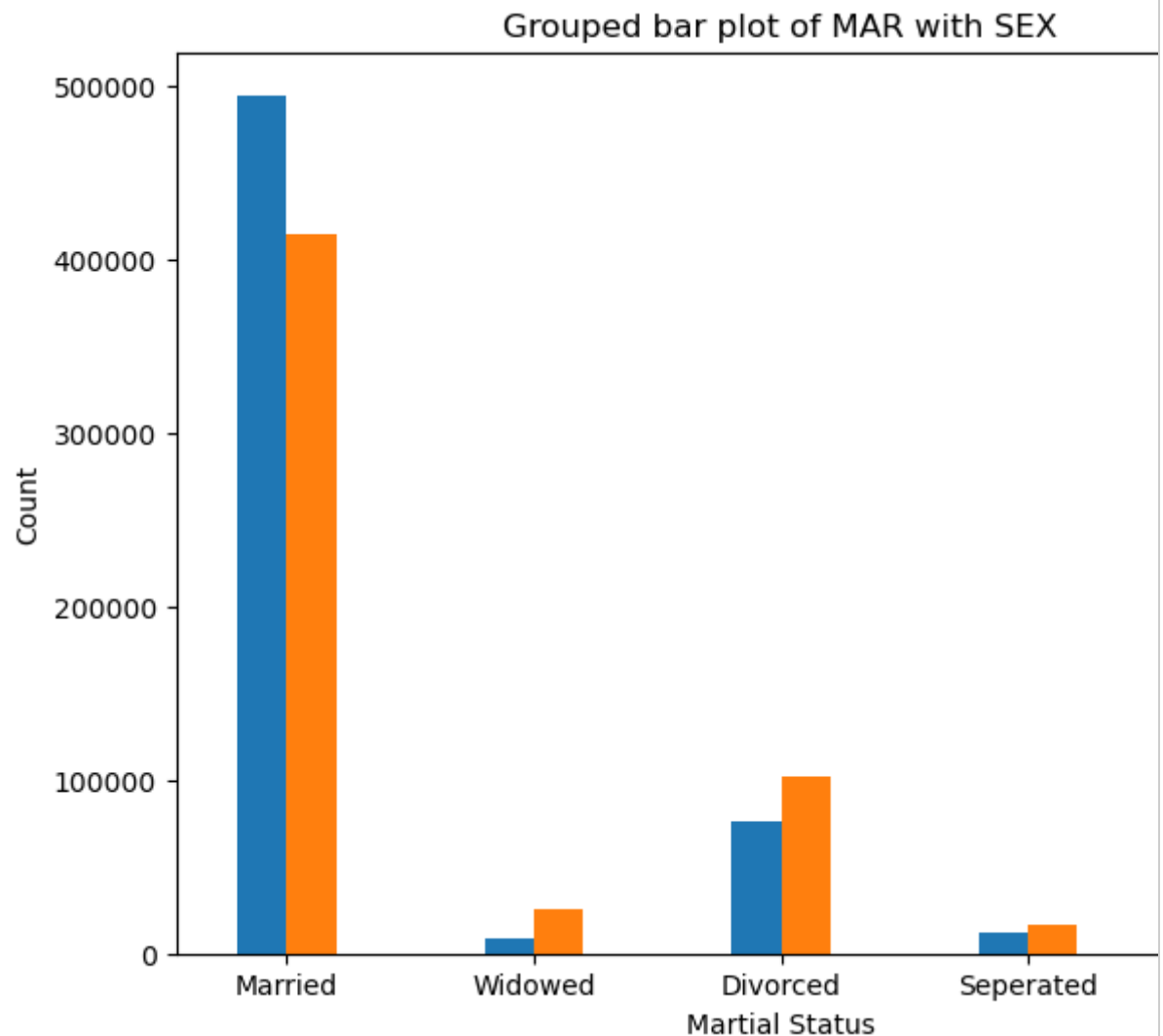
Q5b: Visualize: Amounts: Multiple Variables

Create a grouped bar plot of `MAR` with `SEX`.

In [45]:

```
# Create plot
# Source: https://www.geeksforgeeks.org/create-a-grouped-bar-plot-in-matplotlib/
# Grouping MAR WITH SEX
MAR_WITH_SEX = income.groupby(['MAR', 'SEX']).size().unstack()
fig, ax = plt.subplots(figsize = (8,6))
MAR_WITH_SEX.plot( kind = 'bar', ax = ax, width = 0.4)
plt.xlabel("Marital Status")
plt.ylabel("Count")
plt.title("Grouped bar plot of MAR with SEX")
labels_for_sex = ["Male", "Female"]
```

```
plt.legend(title = "SEX", labels = labels_for_sex)
labels_for_marital_status = ["Married", "Widowed", "Divorced", "Seperated", "Never
Married"]
plt.xticks(range(0,5), labels_for_marital_status, rotation = "horizontal")
plt.show()
```



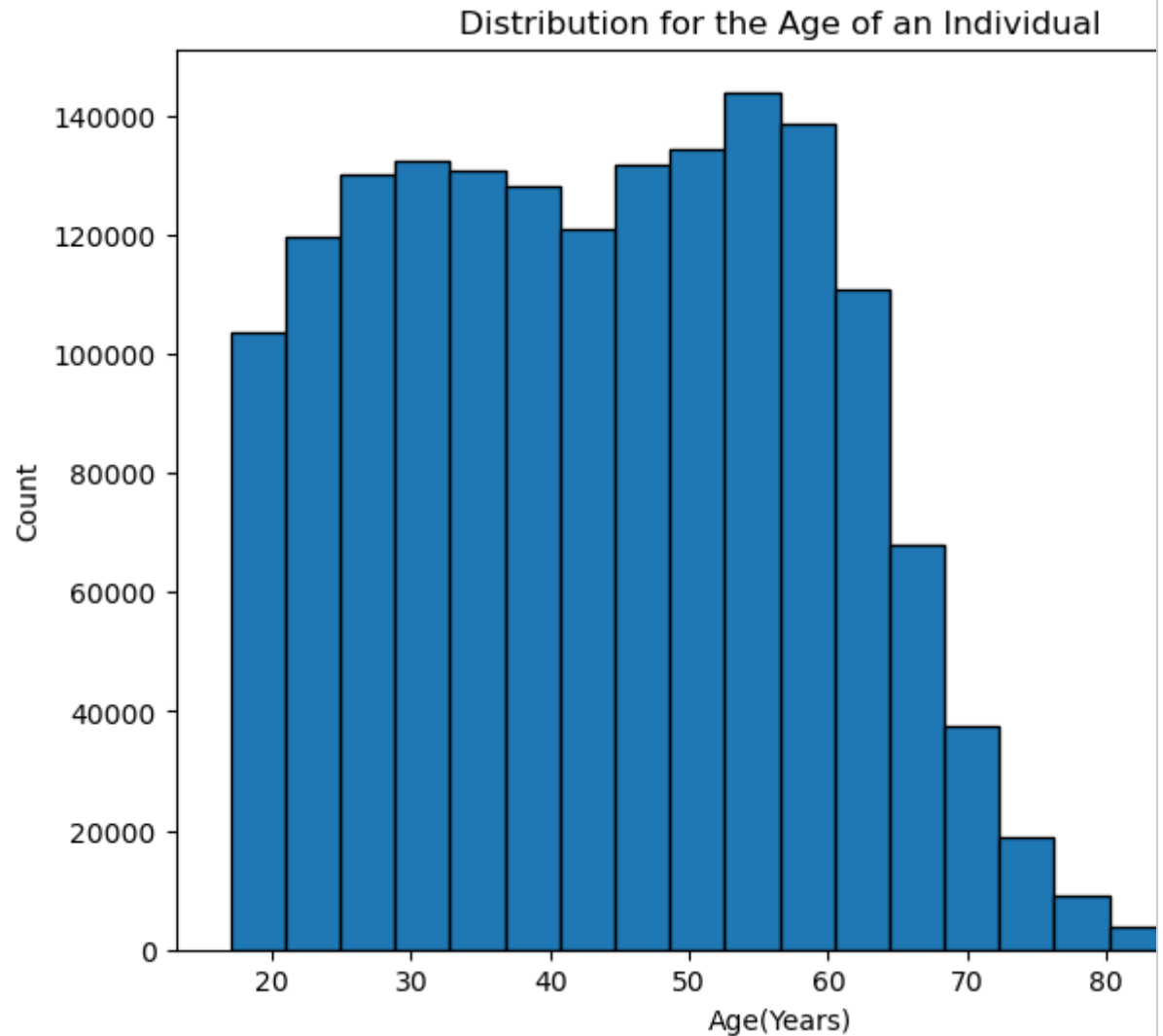
Q5c - Visualize: Distribution: Single Attribute

Generate a histogram with an appropriate number of bins to visualize the distribution of `AGEP`.

In [46]:

```
# Create plot
plt.figure(figsize = (8,6))
plt.hist(income.AGEP, bins = 20, edgecolor = "black")
plt.xlabel("Age(Years)")
plt.ylabel("Count")
plt.title("Distribution for the Age of an Individual")
```

```
plt.show()
```



Q5d - Visualize: Distribution: Multiple Attributes

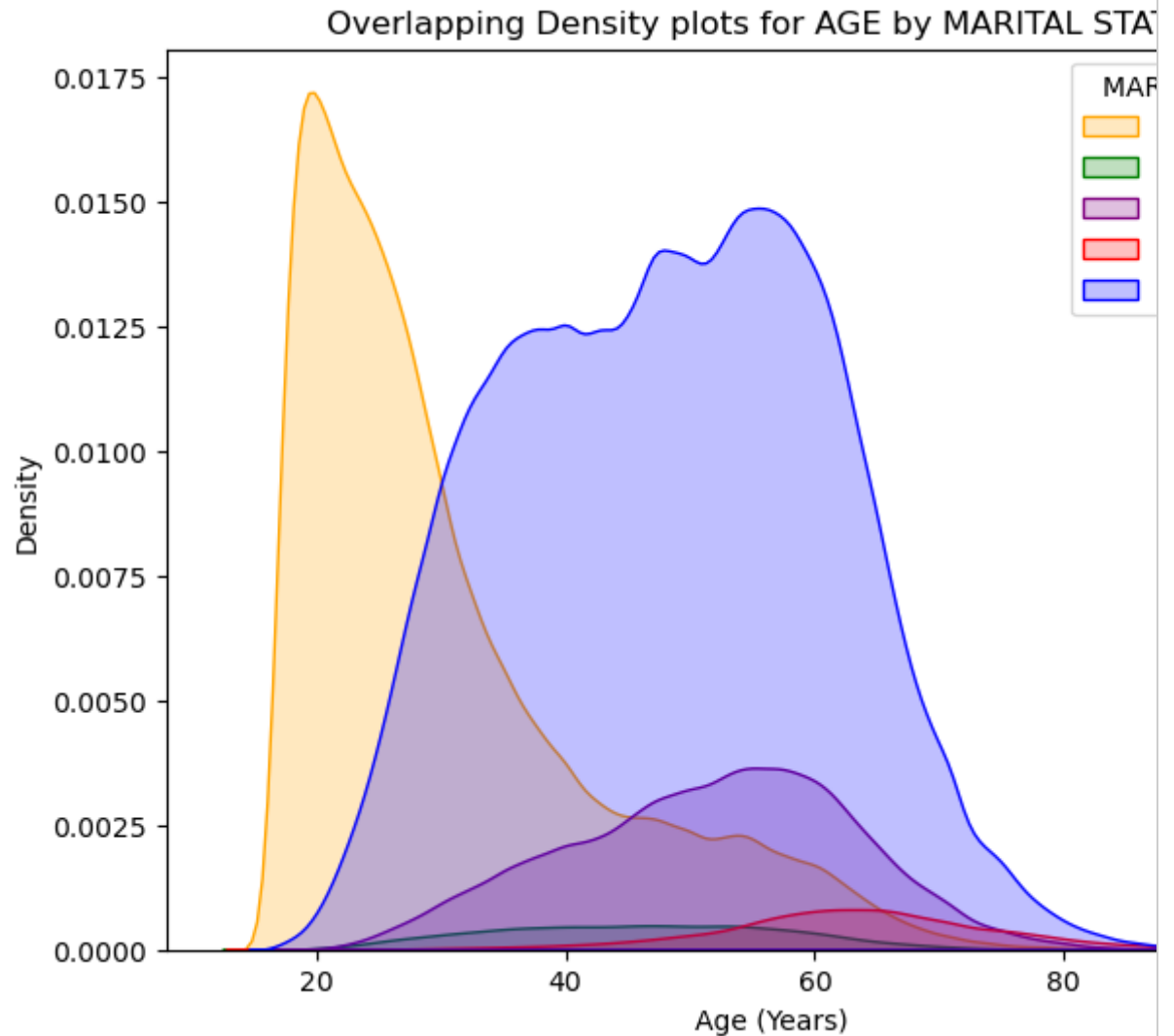
Create an overlapping density plots for `AGEP` by `MAR`.

In [47]:

```
# Create plot
plt.figure(figsize = (8,6))
colors = { 1 : "blue", 2 : "red", 3 : "purple", 4 : "green", 5 : "orange"}
sns.kdeplot( data = income, x = "AGEP", hue = "MAR", fill = True, palette = colors)
plt.xlabel("Age (Years)")
plt.ylabel("Density")
plt.title("Overlapping Density plots for AGE by MARITAL STATUS")
labels_for_marital_status = ["Married", "Widowed", "Divorced", "Separated", "Never
Married"]
plt.legend( title = "MARITAL STATUS", labels = labels_for_marital_status)
```

```
plt.show()
```

/home/campus19/trkosire/.conda/envs/cs5831/lib/python3.10/site-packages/seaborn/_old
with pd.option_context('mode.use_inf_as_na', True):



Submission


Make sure you have run all cells in your notebook in order before running the cell below, so that all images/graphs appear in the output. The cell below will generate a zip file for you to submit. **Please save before exporting!**

NOTE the submission must be run on the campus linux machines. See the instruction in the Canvas assignment.

In []:


```
# Save your notebook first, then run this cell to export your submission.  
grader.export(pdf=False, run_tests=True)
```

▼ .OTTER_LOG

 Download

1	Binary file hidden. You can download it using the button above.
---	-----------------------------------------------------------------

▼ __zip_filename__

 Download

1	a1_2024_02_03T21_58_51_445954.zip
---	-----------------------------------